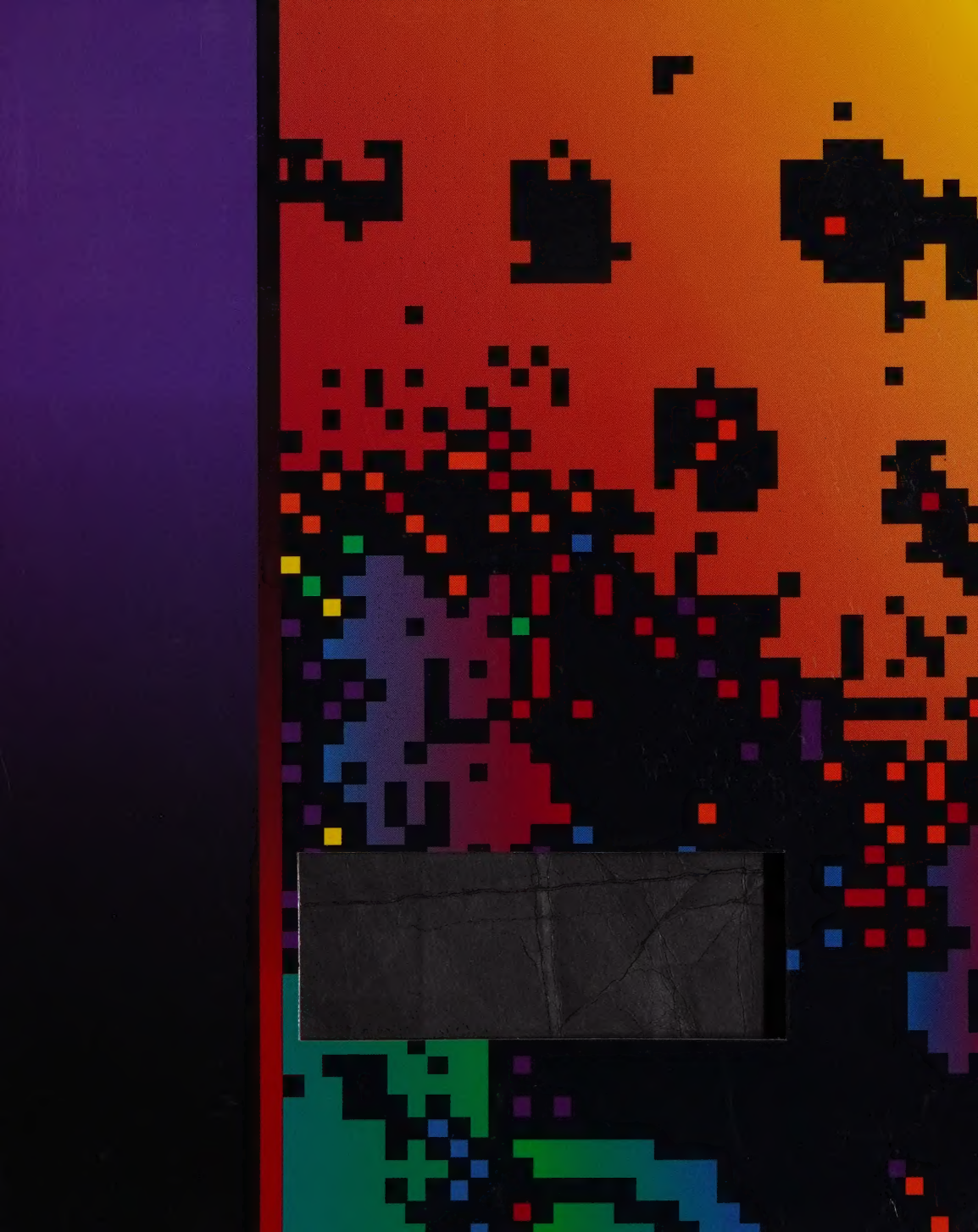


the smart approach to instrumentation

Digital488/80A

Digital I/O Interface User's Manual

DIGITAL/80A-901, rev 2.0 July 1997

IOtech, Inc., 25971 Cannon Road, Cleveland, Ohio 44146

Phone: (216) 439-4091 Fax: (216) 439-4093 e-mail: sales@iotech.com
Internet: http://www.iotech.com

Copyright by IOtech, Inc., Printed in U.S.A

Warranty

IOtech, Inc. warrants our products to be free from defects in material and workmanship. Our obligation under this warranty is to repair or replace any of our products that prove to be defective within two years of shipment. No other warranty is expressed or implied. This warranty is void for products operated in environments or for purposes beyond those described in this manual. IOtech does not guarantee product suitability for all applications.

Limitation of Liability

IOtech cannot be held liable for any damages resulting from the use or misuse of this product.

Copyright & Trademark

All IOtech documentation, software, and circuit boards are copyrighted with all rights reserved by IOtech. No part of this product may be copied, reproduced or transmitted by any mechanical, photographic, electronic, or other method without prior written consent of IOtech, Inc. Digital488/80A and related product names are registered trademarks of IOtech, Inc., 25971 Cannon Road, Cleveland, Ohio 44146. All other product names are trademarks or registered trademarks of their respective holders.

Quality Notice

Certified to ISO 9001 standards since 1996, IOtech applies tests and procedures to assure the highest quality in all aspects of our products. Prior to shipment, we thoroughly test the products and review the documentation. In a spirit of continuous improvement, IOtech welcomes your suggestions.

CE Notice

Many IOtech products now carry the CE marker to indicate they comply with safety and emissions standards as regulated by the European Community. A Declaration of Conformity states which specifications and operating conditions apply.

Safety/Caution Notice

Digital488/80A and related products contain no user-serviceable parts other than expressed in this manual. Refer all service to qualified personnel. Using this equipment in ways other than described in this manual can present serious safety hazards or cause equipment damage.

This warning symbol is used in this manual or on the equipment to warn of possible injury or death from electrical shock under noted conditions.

This caution symbol is used to warn of possible personal injury or equipment damage under noted conditions. Follow all recommendations noted in the manual and the standards of professional practice.

This ESD caution symbol is used to urge proper handling of equipment or components sensitive to damage from electrostatic discharge. Handling guidelines include properly grounded mats and wrist straps, ESD bags and cartons, and related procedures.

Digitized by the Internet Archive
in 2026

https://archive.org/details/manualsplus_11922

Table of Contents

1 Introduction 1.1

1.1 General Description 1.1

1.1.1 I/O Lines 1.1

1.1.2 Handshake/Control Lines 1.1

1.1.3 Addressing 1.1

1.1.4 Software programmable features 1.2

1.1.5 Power-up and setup configuration 1.2

1.1.6 Internal Buffer 1.2

1.1.7 HVCX1 High Current/Voltage Interface Option 1.2

1.2 Available Accessories 1.3

1.3 Specifications* 1.3

1.4 Abbreviations 1.4

2 Getting Started 2.1

2.1 Inspection 2.1

2.2 Hardware Setup 2.2

2.2.1 External Hardware Setup 2.2

2.2.1.1 IEEE 488 Address Selection 2.2

Figure 2.1: S1 Factory Default Settings 2.2

2.2.1.2 Digital488/80A Addressing Modes 2.3

Figure 2.2: IEEE 488 Bus Address Selection (S1) . . . 2.3

Figure 2.3: Secondary Addressing Mode 2.3

Figure 2.4: Dual Primary Addressing Mode (Default) . . 2.3

2.2.2 Internal Hardware Setup 2.4

Figure 2.5: Dual and Secondary Addressing Modes . . . 2.4

2.2.2.1 Operation Voltage 2.5

2.2.2.2 Configuring the HVCX1 Option 2.5

Figure 2.6: HCVX1 Option 5, 12, 24 or 48 V Logic Input 2.6

Figure 2.7: HVCX1 Option - High Current/Voltage Output 2.6

2.2.2.2.1 Setting up Input Ports 2.7

Figure 2.8: Channel 1 Input Setup 2.8

Figure 2.9: Channel 0 Input Setup 2.8

2.2.2.2.2 Setting Up Handshake Input Lines 2.9

2.2.2.2.3 Setting Up Output Ports 2.9
Figure 2.10: Handshake Input Line Setup 2.9
Figure 2.11: Channel 1 Output Setup 2.10
Figure 2.12: Channel 0 Output Setup 2.10
2.2.2.2.4 Setting Up Handshaking Output Lines . . . 2.11
2.2.2.2.5 Installing HVCX1 . . . . . 2.11
Figure 2.13: Handshake Output Lines Set for TTL . . . 2.11
Figure 2.14: Handshake Output Lines - User Defined . 2.11
2.2.3 Mounting 2.12
2.2.3.1 Rack Mount 2.12
2.2.3.2 Bench Top 2.12
Figure 2.15: Enclosure Side View 2.12
Figure 2.16: Rack Mounting Hardware (Top View) . . . 2.12
2.2.4 Power Hookup; Power On 2.13
2.3 IEEE 488 Bus Output Terminator Selection 2.13
2.4 Front Panel Indicators 2.13
2.5 Programming Examples 2.14
Figure 2.17: Front Panel Indicator Lights 2.14
3 Additional Information 3.1
3.1 Handshaking/Control Lines 3.1
3.1.1 Clear (Pin 41) 3.1
Figure 3.1: Timing Diagram for Clear Output 3.1
3.1.2 Data Strobe (Pin 42) 3.2
3.1.3 External Data Ready [EDR] (46) 3.2
Figure 3.2: Timing Diagram for Strobe Output 3.2
3.1.4 Inhibit (Pin 44) 3.3
3.1.5 Trigger (Pin 43) 3.3
Figure 3.3: EDR Input & Inhibit Output Timing Diagram 3.3
Figure 3.4: Timing Diagram for Trigger Output 3.3
3.1.6 Service (Pin 45) 3.4
3.2 IEEE 488 Bus Implementation 3.4
3.2.1 My Talk Address (MTA) 3.4

3.2.2 My Listen Address (MLA) 3.5
3.2.3 Device Clear (DCL and SDC) 3.5
3.2.4 Group Execute Trigger (GET) 3.5
3.2.5 Interface Clear (IFC) 3.5
3.2.6 Serial Poll Enable (SPE) 3.6
3.2.7 Serial Poll Disable (SPD) 3.6
3.2.8 Unlisten (UNL) 3.6
3.2.9 Untalk (UNT) 3.6
3.2.10 Serial Poll Response 3.6
3.2.11 Data Transfer Speed Specifications 3.7
3.2.12 Collection and Storage of Data 3.7
3.2.13 Output of Data to Ports 3.7
3.2.14 IEEE 488 Bus Communication 3.8
3.2.15 Logic Levels - Base Unit 3.8
3.3 Digital Input/Output Ports 3.8
3.3.1 Using the HVCX1 Option for High Voltage/High Current Outputs 3.8
3.4 Digital I/O Port Pin Outs 3.9
Figure 3.5: DB-50 Using High Level Outputs 3.9
Figure 3.6: DB-50 Rear Panel Connector Pin-Outs . . . 3.9
Table 3.1: Digital488/80A Pinouts 3.10
Figure 3.7: HVCX1 Option Block Diagram 3.11
Figure 3.8: HVCX1 Channel Block Diagram (40 I/O Lines) 3.11
Figure 3.9: Typical HVCX1 Port Schematic 3.12
4 Command Descriptions 4.1
Bit Set (An) 4.2
Bit Clear (Bn) 4.3
Configure (Cn) 4.4
Data (Dn...Z) 4.5
Error (E?) 4.7
Format (Fn) 4.9
Bus Input/Output (Gn) 4.16
Handshake (Hn) 4.17

Invert (In) 4.18
End or Identify (EOI) (Kn) 4.19
Buffer (Ln) 4.20
Service Request Mask (SRQ) (Mn) 4.21
Recall Configuration (On) 4.23
Port (Pn) 4.24
Inhibit (Qn) 4.25
Data Ready (Rn) 4.26
Save Configuration (Sn) 4.28
Test (Tn) 4.29
Status (Un) 4.30
View Configuration (Vn) 4.34
View Version (V?) 4.35
Execute (X) 4.36
Terminator (Yn) 4.37
Query (?) 4.38
Serial Poll Status Byte () 4.39

5 IEEE 488 Primer 5.1

5.1 History 5.1
5.2 General Structure 5.1
5.3 Send It To My Address 5.2
5.4 Bus Management Lines 5.2
5.4.1 Attention (ATN) 5.2
5.4.2 Interface Clear (IFC) 5.3
5.4.3 Remote Enable (REN) 5.3
5.4.4 End or Identify (EOI) 5.3
Figure 5.1: IEEE 488 Bus Structure 5.3
5.4.5 Service Request (SRQ) 5.4
5.5 Handshake Lines 5.4
5.5.1 Data Valid (DAV) 5.4
5.5.2 Not Ready for Data (NRFD) 5.4
5.5.3 Not Data Accepted (NDAC) 5.4
5.6 Data Lines 5.5

5.7 Multiline Commands 5.5
5.7.1 Go To Local (GTL) 5.5
5.7.2 Listen Address Group (LAG) 5.5
5.7.3 Unlisten (UNL) 5.5
Figure 5.2: IEEE 488 Bus Handshaking 5.5
5.7.4 Talk Address Group (TAG) 5.6
5.7.5 Untalk (UNT) 5.6
5.7.6 Local Lockout (LLO) 5.6
5.7.7 Device Clear (DCL) 5.6
5.7.8 Selected Device Clear (SDC) 5.6
5.7.9 Serial Poll Disable (SPD) 5.6
5.7.10 Serial Poll Enable (SPE) 5.6
5.7.11 Group Execute Trigger (GET) 5.6
5.7.12 Take Control (TCT) 5.7
5.7.13 Secondary Command Group (SCG) 5.7
5.7.14 Parallel Poll Configure (PPC) 5.7
5.7.15 Parallel Poll Unconfigure (PPU) 5.7
5.8 More on Service Requests 5.7
5.8.1 Serial Poll 5.8
5.8.2 Parallel Poll 5.8
6 Service Information 6.1
6.1 Factory Service 6.1
6.2 Theory of Operation 6.1
6.3 Digital488/80A Motherboard Component Layout 6.2
6.4 Digital488/80A Motherboard Replaceable Parts List 6.3
6.5 HVCX1 Option Component Layout 6.4
6.6 HVCX1 Option Replaceable Parts List 6.5
A Appendix A: Digital 488/80A Command Summary A.1
B Appendix B: Keyboard Controller Program B.1
C Appendix C: Character Codes and IEEE Multiline Messages . . C.1

Introduction

1.1 General Description

The Digital488/80A is a digital input and output interface to the IEEE 488 bus that enables an IEEE 488 controller to program 80 bits of TTL level signals as either inputs or outputs. It includes six handshake/control lines for each 40-bit channel and two addressing modes. The unit is fully programmable, including the power-up configuration, which is stored in non-volatile RAM. The internal data buffer in the Digital488/80A can store up to 2,000 40-byte words. With the addition of the HVCX1 High Current/High Voltage Interface option, the I/O lines can be configured as high voltage inputs or as high current/high voltage outputs to allow interfacing with solenoids, switching relays and other high voltage/high current devices.

1.1.1 I/O Lines

The Digital488/80A has 80 TTL level digital I/O lines that are divided into two 40-bit channels. Each channel is divided into five eight-bit ports. Six high level ASCII commands configure the ports to be all inputs, all outputs or combinations in between. Thus, a configuration can range from 80 input lines to 80 output lines or combinations in between in multiples of 8 bits.

1.1.2 Handshake/Control Lines

Each 40-bit channel has six handshake/control lines. The Trigger output signal is pulsed when a Group Execute Trigger (GET) command is received on the bus. The Service line is an edge-triggered input that generates a Service Request on the bus. The Data Strobe output is pulsed when the IEEE 488 controller presents new data on the I/O lines. The Inhibit output line is asserted while the IEEE 488 controller is reading data from input lines. The Clear output line is pulsed whenever a Device Clear or Selected Device Clear command is received from the IEEE 488 controller. The External Data Ready input line is used to latch digital input data on the I/O lines.

1.1.3 Addressing

Two addressing modes are available in the Digital488/80A. Dual primary addressing permits two channels to reside at two consecutive primary bus addresses. Secondary addressing allows two channels to reside at the same primary bus address with two consecutive secondary addresses.

When addressed to talk, the Digital488/80A outputs data from all forty bits of the selected channel or a selected eight bit port. When addressed to listen, the unit accepts input data

and programming information from the controller and outputs the data to the appropriate I/O port.

1.1.4 Software programmable features

The software programmable features on the Digital488/80A include data formats, individual bit set and bit clear commands and terminators. The six available data formats are ASCII hexadecimal, ASCII character, ASCII binary, binary, high speed binary and ASCII decimal. The programmable terminators facilitate interfacing to various controllers. Using the Status command, the controller can check the programmed status of the Digital488/80A at any time. With the Query (?) option offered with each Digital488/80A command, user-defined status command strings can be built to request the status information needed for a given application.

1.1.5 Power-up and setup configuration

The Digital488/80A allows up to 100 configurations for each channel to be created. The configurations are stored in non-volatile memory, which enables the input/output status and logic level of every I/O line to be defined when power is applied. The configurations can be called from a single pre-defined command from the IEEE 488 controller. This allows easy configuration of the device for a variety of applications.

Internal pull-up resistors ensure a logic "high" the instant power is applied. When using the HVCX1 option with high current driver outputs, the internal pull-up resistors ensure that the high current driver outputs are switched off the instant power is applied - a requirement for process control applications.

1.1.6 Internal Buffer

A built-in data buffer in the Digital488/80A can store up to 8,000 bytes, which is large enough to capture two thousand patterns from one 40-bit I/O channel. When interfacing to slow devices, the data buffer relieves the controller of the need to constantly read data from the Digital488/80A.

1.1.7 HVCX1 High Current/Voltage Interface Option

The HVCX1 option permits the I/O lines to be configured as high voltage (up to 50 V) inputs or as high current/high voltage outputs (up to 200 mA sink capability) to allow interfacing with solenoids, switching relays and other high voltage/high current devices using an external voltage source. It allows interfacing to non-TTL level devices by offering compatibility with 12, 24 and 48 volt logic. The individual 8-bit ports can be configured to control high current open collector drivers. Each high current driver has an internal fly-back diode for protection when switching inductive devices such as relays. When the HVCX1 option is used, the six handshake and control lines can also interface to 12, 24, or 48 volt logic.

1.2 Available Accessories

Additional accessories that can be ordered for the Digital488/80A include:

CA-7-1	1.5 foot IEEE 488 cable
CA-7-2	6 foot IEEE 488 cable
CA-7-3	6 foot shielded IEEE 488 cable
CA-7-4	6 foot reverse entry IEEE 488 cable
CA-88	6 foot 50-pin D-shell ribbon cable
CN-20	Right angle IEEE 488 adapter, male and female
CN-22	IEEE 488 multi-tap bus strip, four female connectors in parallel
CN-23	IEEE 488 panel mount feed-through connector, male and female

1.3 Specifications

CAUTION

Please read this manual carefully! If equipment is used in any manner not specified in this manual, the protection provided by the equipment may be impaired.

Digital I/O

Terminal Installation Category:

- **Standard:** Not Applicable.
- **CE:** Category 1.

Digital I/O Capability:

- Dual Channels, each with five 8-bit ports, programmable as inputs or outputs. Also included per channel are programmable handshake lines, data latching capability, a Service Request (SRQ) input and Clear and Trigger outputs.

Logic Levels:

- Outputs drive 2 TTL loads (3.2 mA). Optional 12, 24, 48 volt logic support available with the HVCXI High Current / Voltage Interface option.

High-Current Outputs:

- With the HVCX1 option, the outputs can be configured, in groups of eight, as high current outputs. Each high-current output is capable of sinking up to 200 mA at 50 VDC maximum. Total current sinking capability is 8 A maximum.

Update Rate:

- In binary mode, the Digital488/80A can update one 40 bit I/O channel at over 2K bytes per second.

I/O Port Connections:

- Two 50-pin male D-shell connectors. Mating solder tab connectors are supplied.

Input Ranges, Nominal Logic Low/High Voltages (±10%), and Input Resistances (HVCX1 option only):

- 0-5 volt range; 0.8 volts low / 2.4 volts high; 15K Ω
- 0-12 volt range; 1.9 volts low / 5.75 volts high; 35K Ω
- 0-24 volt range; 4.2 volts low / 11.5 volts high; 61K Ω
- 0-48 volt range; 8.4 volts low / 23 volts high; 135K Ω

IEEE 488 Interface

CAUTION

The IEEE 488 terminal must only be used to control a non-isolated IEEE 488 system. The common mode voltage (cable shell to earth) must be zero.

Terminal Installation Category:

- **Standard:** Not Applicable.
- **CE:** Category 1.

Implementation:

- SH1, AH1, T4, TE8, L4, LE4, SR1, RL0, PP0, DC1, DT1, C0, E1.

Terminators:

- Ignored on input. Selectable CR, LF, LF-CR or CR-LF, with or without EOI.

Programmable:

- IEEE Terminators, EOI, SRQ Mask, Port Data, Active Levels, Handshake Lines, Format and Configuration.

Connectors:

- Standard IEEE 488 connector with metric studs.

General

WARNING

Service: This product contains no operator serviceable parts. Service must be performed by qualified personnel. Never remove the cover while the AC line voltage is connected. Internal voltage potentials exist which could cause bodily injury or death.

Terminal Installation Category:

- **Standard:** Not Applicable.
- **CE:** Category 2 for Line Voltage Input terminal. Category 1 for all other terminals.

Dimensions:

- 425 mm wide x 45 mm high x 203 mm deep (16.75" x 1.75" x 8")

Weight:

- 2 kg (4.5 lbs.)

Operating Environment:

- **Standard:** Indoor use, 0 to 50°C, 0 to 95% RH non-condensing
- **CE:** Indoor use at altitudes below 2000 m, 0 to 40°C, 80% maximum RH up to 31°C decreasing linearly 4% RH/°C to 40°C.

Controls:

- Power Switch, external DIP switch for IEEE 488 addressing mode and IEEE 488 address.

Indicators:

- LED indicators for TALK, LISTEN, SRQ, ERROR, TEST and POWER.

Power:

- 105-125 VAC, or 210-250 VAC, 50/60 Hz; 30 VA maximum.

CAUTION

Line Voltage: The protective conductor terminal on the AC line connector must be connected to an external protective earthing system. Failure to make such a connection will impair protection from shock.

Fuse:

- 1/4A Slo Blo (for 105-125V power line), or 1/8A Slo Blo (for 210-250V power line).

CAUTION

Fuse Failure: Fuse failure indicates a possible problem within the device circuitry. If a fuse blows, contact a qualified service representative. Replacement fuses are to be installed by qualified service personnel with the unit disconnected from the power source and with all other terminals disconnected. If the line voltage selector is changed, the fuse designated for use with that line voltage must be used.

1.4 Abbreviations

The following IEEE 488 abbreviations are used throughout this manual.

addr n	IEEE bus address "n"	PPC	Parallel Poll Configure
ATN	Attention line	PPU	Parallel Poll Unconfigure
CA	Controller Active	SC	System Controller
CR	Carriage Return	SDC	Selected Device Clear
data	Data String	SPD	Serial Poll Disable
DCL	Device Clear	SPE	Serial Poll Enable
GET	Group Execute Trigger	SRQ	Service Request
GTL	Go To Local	TA	Talker Active
LA	Listener Active	TAD	Talker Address
LAG	Listen Address Group	TCT	Take Control
LF	Line Feed	term	Terminator
LLO	Local Lock Out	UNL	Unlisten
MLA	My Listen Address	UNT	Untalk
MTA	My Talk Address	*	Unasserted

Getting Started

2.1 Inspection

The Digital488/80A was carefully inspected, both physically and electronically, prior to shipment. When you receive the unit, carefully unpack all items from the shipping carton and check for any obvious signs of physical damage which may have occurred during shipment. Promptly report any such damage to the shipping agent. Remember to retain all shipping materials in the event that return shipment becomes necessary. Every Digital488/80A is shipped with the following items:

- Digital488/80A — IEEE 488 Digital I/O Converter
- Digital/80A-901 — Digital488/80A User's Manual
- CA-1 — Power Cable
- HA-41-6 — Screws (4)
- FE-1 — Rubber Feet
- EN-6 — Rack Ears (2)
- FU-1-.25 — 1/4 Amp Replacement Fuse
- 196-4030 — Jumper Board
- CN-18-50 — Mating Connector for DB-50 (2)

In addition, if the High Voltage/Current Interface Option (HVCX1) is included, you will have the following items:

- HVCX1 — High Voltage/Current Interface Option
- RN-3-20K — 20KΩ Resistor DIP Pack (12)
- RN-3-56K — 56KΩ Resistor DIP Pack (12)
- RN-3-120K — 120KΩ Resistor DIP Pack (12)
- RN-9-27K — 27KΩ Pullup Resistor SIP Pack (12)

2.2 Hardware Setup

WARNING

The Digital488/80A is intended for INDOOR USE ONLY. Failure to observe this warning could result in equipment failure, personal injury or death.

2.2.1 External Hardware Setup

The Digital488/80A has a DIP switch (S1) which is accessible from the unit's rear panel. The position of S1's micro-switches determines the IEEE 488 address and addressing mode. Switch S1 is read only during power up, and for this reason **must have the proper settings prior to applying power.** The factory default setting of S1 is for a IEEE Bus Address of 8, and for the "Dual Primary Addressing Mode." In the dual primary addressing mode micro-switches 6 and 7 are ignored.

Note: Switch S1 is read only during power up. For this reason you must make any setting changes prior to applying power to the unit.

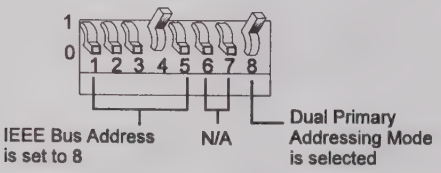


Figure 2.1: S1 Factory Default Settings

To modify any of these defaults, disconnect the power cord from the power supply and change the switch settings using a small screwdriver. The enclosure does not need to be opened to change S1 switch settings.

2.2.1.1 IEEE 488 Address Selection

The IEEE 488 bus address is set by positioning S1's micro-switches 1 through 5 (located on rear of unit). The address can be set from 0 through 30 and is read only at power on. The address is selected by simple binary weighting, with micro-switch 1 being the least significant bit (with a value of 1) and micro-switch 5 being the most significant bit (with a value of 16). The factory default is address 8.

Note: If address 31 is selected, there is an automatic default to address 30. This is because the IEEE 488 standard has reserved address 31 for Unlisten Device.

Note: The Digital488/80A bus address must agree with the IEEE 488 controller address.

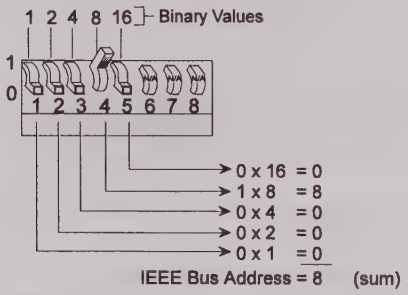


Figure 2.2: IEEE 488 Bus Address Selection on S1

2.2.1.2 Digital488/80A Addressing Modes

The Digital488/80A addressing mode is determined by S1's micro-switch 8, accessible from the rear of the unit. With micro-switch 8 in the up position, "Dual Primary Addressing Mode" is selected; with the switch in the down position, "Secondary Addressing Mode" is selected. The switch settings for both modes are shown in the following two figures. A detailed description of dual primary and secondary addressing follows.

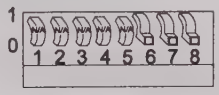


Figure 2.3: Secondary Addressing Mode

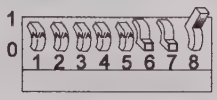


Figure 2.4: Dual Primary Addressing Mode

The Digital488/80A can be thought of as two identical IEEE 488-to-digital I/O interfaces. Each interface occupies one bus address and has one I/O channel. For this reason the Digital488/80A occupies two bus addresses in an IEEE 488 system.

The default addressing on the Digital488/80A is "dual primary," in which two consecutive bus addresses are used. Dual primary addressing offers ease of use at the expense of two bus address locations. This may not be practical for applications in which the bus address locations are required by other equipment. In these situations, "secondary addressing" may be used. Secondary addressing uses a single bus address for multiple interfaces.

Dual Primary Addressing. When dual primary addressing is selected (micro-switch 8 in the up position) , S1's micro-switch 1 (the least significant bit of the address) is ignored. The Channel 0 interface resides at an even numbered bus address, while the Channel 1 interface resides at the next higher odd address. For example, if the address switches are set for bus address 8, Channel 0's interface resides at bus address 8 and Channel 1's interface resides at bus address 9 (as indicated in Figure 2.6).

Note: If address 30 is selected when in dual primary mode, the Digital488/80A defaults to address 28 for Channel 0, and to address 29 for Channel 1.

Secondary Addressing. When secondary addressing is used (micro-switch 8 in the down position), up to four Digital488/80A units can reside at the *same primary bus address*. The secondary addresses (at which Channel 0 and Channel 1 reside) is selected with micro-switches 6 and 7, as indicated in Figure 2.5.

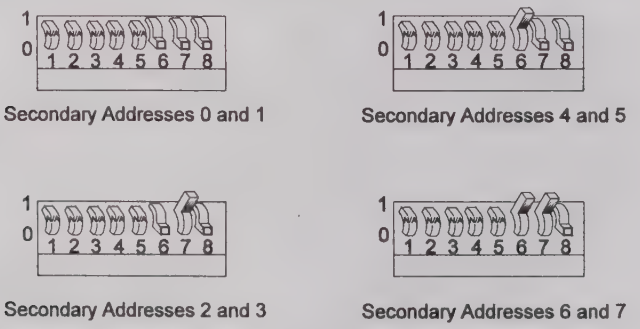

1
0
1 2 3 4 5 6 7 8
Secondary Addresses 0 and 1
Secondary Addresses 4 and 5
Secondary Addresses 2 and 3
Secondary Addresses 6 and 7

Figure 2.5: Secondary Addressing Configurations

For example, if two Dital488/80A units are configured for primary address 08, with the first unit being configured for secondary addresses "0 and 1", and the second unit configured for secondary addresses "1 and 2", then the communication is as indicated by the following table.

Communication Example			
To Communicate With:		**Use Primary Address:**	**Use Secondary Address:**
Unit 1	Channel 0	08	00
	Channel 1	08	01
Unit 2	Channel 0	08	02
	Channel 1	08	03

Note: When in the secondary address mode, if IEEE bus address 31 is selected, the Digital488/80A unit will automatically default to primary address 30.

Figure 2.6 shows serves as a comparison between the two addressing modes.

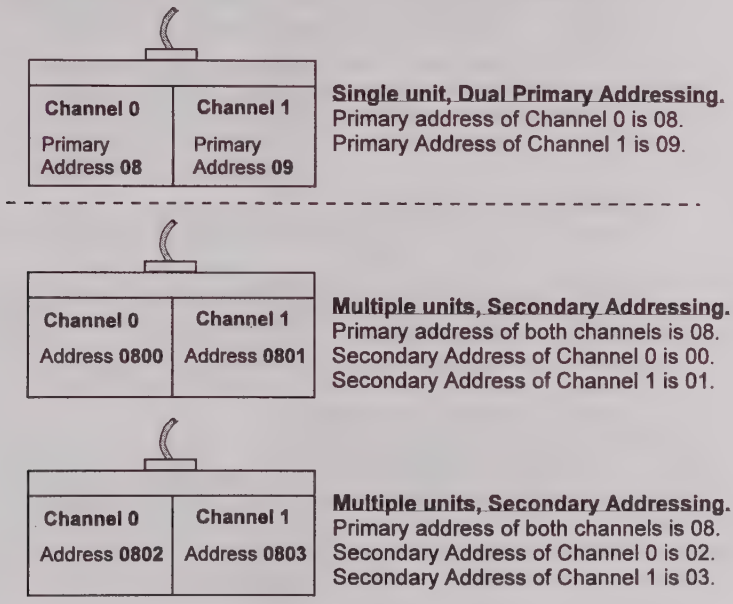

Channel 0
Primary Address 08
Channel 1
Primary Address 09
Single unit, Dual Primary Addressing.
Primary address of Channel 0 is 08.
Primary Address of Channel 1 is 09.
Channel 0
Address 0800
Channel 1
Address 0801
Multiple units, Secondary Addressing.
Primary address of both channels is 08.
Secondary Address of Channel 0 is 00.
Secondary Address of Channel 1 is 01.
Channel 0
Address 0802
Channel 1
Address 0803
Multiple units, Secondary Addressing.
Primary address of both channels is 08.
Secondary Address of Channel 0 is 02.
Secondary Address of Channel 1 is 03.

Figure 2.6: Dual Primary Addressing & Secondary Addressing Modes, A Comparison

2.2.2 Internal Hardware Setup

WARNING

Never open the Digital488/80A casing while the unit is connected to power. Internal voltage potentials exist which could cause personal injury or death.

Internal hardware setup consists of choosing 110 Vac or 220 Vac power and installing the optional High Voltage/Current Interface option (HVCX1) if applicable.

Before accessing the internal hardware, disconnect the power cord from the power line and from the Digital488/80A. Disconnect IEEE 488 and digital I/O cables prior to disassembling the unit.

To open the unit place it on a flat surface, then remove the four screws from the top of the case and remove the top cover. After making the required changes, replace the cover; then replace and tighten the four cover screws.

2.2.2.1 Operation Voltage

The operating voltage is set by internal switch S4, located next to the power supply transformer. The Digital488/80A was shipped from the factory set for the operating voltage marked on the label placed over the power jack on the rear panel. If this setting needs to be changed, insert the tip of a small screwdriver into the slot of the switch and move the switch to the left or right so the desired line voltage selection appears on the switch. If the operation voltage is changed, the fuse below S4 (F1 Linefuse) must also be changed. The fuse value for 110 V ac operation is 0.5 A; the value for 220 V ac operation is 0.25 A. Make note of the new settings for later reference.

2.2.2.2 Configuring the HVCX1 Option

The HVCX1 High Current/Voltage Interface option is a configurable daughter board that allows the Digital488/80A to be used with solenoids, switching relays and other high voltage/high current devices. When using the HVCX1 with an external voltage source, the I/O lines may be configured as high voltage inputs (up to 50 V) or as high current/high voltage outputs. The factory default is all 5 V I/O and handshaking for both channels with all ports set as inputs.

When the HVCX1 option is used, the data lines of both channels can be configured as high voltage/high current outputs. Outputs can support up to 50 V dc @ 200 mA not to exceed 8 A total. When the outputs are configured as high voltage, using internal pull-up resistors (i.e., the 27K SIP), all 40 output lines on a channel have their pull-ups and integral flyback diodes pulled up to the flyback pin (pin 48) on the DB-50 connector. When used in this mode, outputs are configured on a per-channel basis. This flyback pin should be connected to the positive supply lead of the power supply used with the external devices. The flyback diode supplies protection against inductive transients.

Pin 48 sets the pull-up logic level for all output lines on a channel. Lines configured with the pull-up can only be used as outputs. These outputs can sink up to 200 mA in the ON state and withstand voltages up to 50 V dc through the use of open collector drivers with integral flyback diodes for inductive load transient suppression.

Inputs are configured on a port-by-port basis (in groups of 8 bits). Inputs can be configured to be compatible with 5, 12, 24 and 48 volt logic. The interface voltage for each port may be set independently from other ports.

See Specifications for threshold levels and other information.

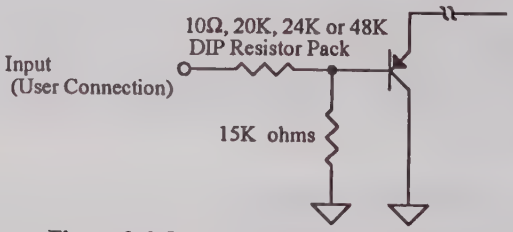


Figure 2.6: HCVX1 Option 5, 12, 24 or 48 V Logic Input

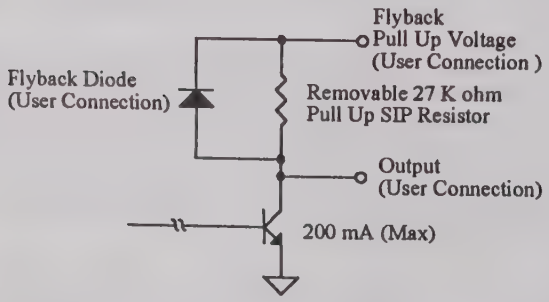


Figure 2.7: HVCX1 Option - High Current/Voltage Output

Before configuring the HVCX1 option, determine which ports are going to be set up for inputs and which for outputs. Determine what voltages are desired for the inputs: 0-5 V, 0-12 V, 0-24 V or 0-48 V. The chart on the next page shows which combinations of outputs and inputs are permitted (limitations set by the Configure software command).

Configure Command Code	Port 5	Port 4	Port 3	Port 2	Port 1
C0	input	input	input	input	input
C1	input	input	input	input	output
C2	input	input	input	output	output
C3	input	input	output	output	output
C4	input	output	output	output	output
C5	output	output	output	output	output

If no chips are installed in the input or output side of a port, the Digital488/80A lines are pulled up and the user will receive "FF" from that port.

2.2.2.2.1 Setting up Input Ports

All of the components mentioned below are for the Port section of the HVCX1 board labeled INPUT.

1. Place the 2596 IC chip into the IC input socket of the port being configured. The notched end of the IC must be to the left (see Figures 2.8 and 2.9 for the proper orientation).
2. Install the DIP (dual in-line package) pack for the voltage desired. See the chart below for the DIP pack labeling. The notched end of the DIP pack must be to the left (see Figures 2.8 and 2.9 for the proper orientation).

Inputs	DIP Resistor Values	Label of DIP Pack Supplied with HVCX1 Option
0-5 V	10 ohm	4116R-001-100
0-12 V	20K	4116R-001-203
0-24 V	56K	4116R-001-563
0-48 V	120K	4116R-001-124

3. Verify that no components (the 2596 and SIP pack) are in the output section of a port being configured as an input. Having components for the output and input installed in the same port will cause damage to the unit.

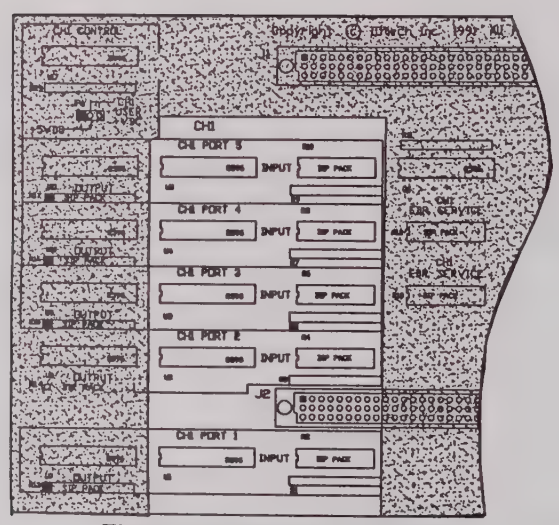


Figure 2.8: Channel 1 Input Setup

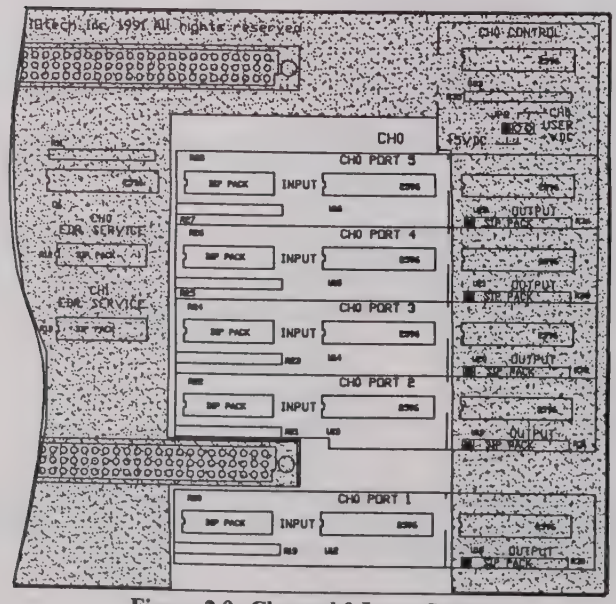


Figure 2.9: Channel 0 Input Setup

2.2.2.2.2 Setting Up Handshake Input Lines

The Input Handshaking voltage ranges are set by resistor DIP packs. The available voltage ranges and DIP packs are:

Inputs	DIP Resistor Value	DIP Pack Label
0-5 V	10 ohm	4116R-001-100
0-12 V	20K	4116R-001-203
0-24 V	56K	4116R-001-563
0-48 V	120K	4116R-001-124

Select the appropriate DIP pack and place it in the sockets in the center of the board labeled `CH0 EDR, SERVICE` for the channel 0 settings; in the section labeled `CH1 EDR, SERVICE` for the Channel 1 settings.

Figure 2.10: Handshake Input Line Setup

2.2.2.2.3 Setting Up Output Ports

All of the components mentioned below are for the Port section labeled `OUTPUT`.

1. Place the 2596 IC chip into the IC output socket of the port being configured. The notched end of the IC must be to the left (see Figures 2.11 and 2.12 for the proper orientation).
2. When configuring the option for high voltage, install the 27K SIP (single in-line package) pull-up resistor network (labeled 9X-1-273) in the SIP socket in the output section. Make sure pin 1 of the resistor SIP pack (denoted by a dot) is toward the left. This pin must be connected to an external voltage supply in order to achieve the desired logic level.
3. Verify that no components (the 2596 and DIP pack) are in the input section of a port being configured as an input. Having components for the output and input installed in the same port will cause damage to the unit.

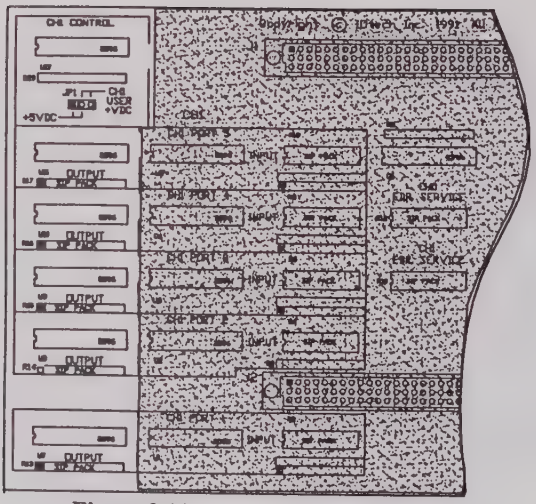


Figure 2.11: Channel 1 Output Setup

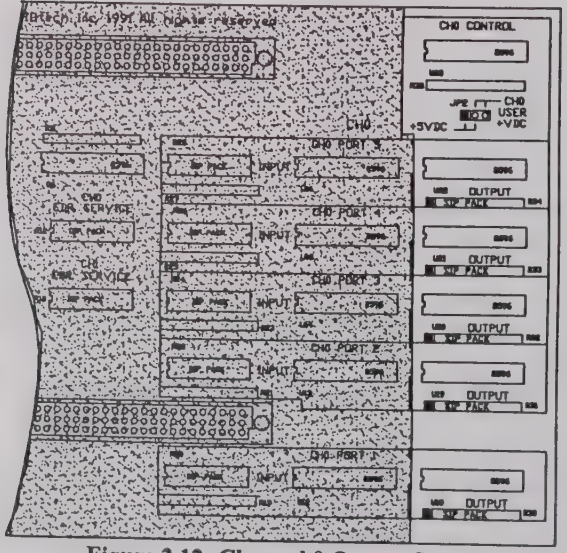


Figure 2.12: Channel 0 Output Setup

2.2.2.2.4 Setting Up Handshaking Output Lines

With the HVCX1 option, the Digital488/80A handshaking outputs (Trigger, Inhibit, Strobe and Clear) can be set for 5 V (TTL) or for a user defined voltage. The user-defined voltage is the voltage supplied through pin 48 of the channel (voltage is the same as the data lines for ports configured as outputs). Handshaking output lines are changed as a group and cannot have their sense reconfigured - handshake inputs are always inputs; handshake outputs are always outputs. There is one group of handshaking output lines per channel (i.e., CHO has handshake set and CH1 has handshake set). Handshaking may be done at TTL levels even if digital I/O is set for a higher range.

The handshake outputs are set by jumpers JP1 and JP2 in the CH0 Control and CH1 Control sections of the HVCX1 option board. See Figures 2.13 and 2.14 for jumper settings.

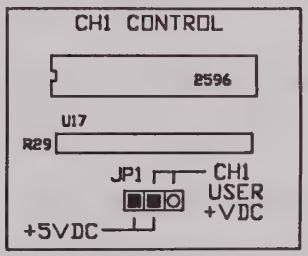


Figure 2.13: Handshake Output Lines Set for TTL

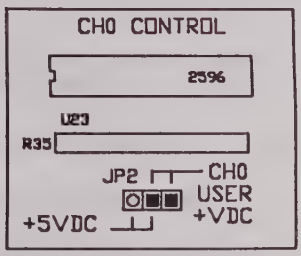


Figure 2.14: Handshake Output Lines - User Defined

2.2.2.2.5 Installing HVCX1

When all configuration is complete, record the settings on the configuration record supplied in the jacket of this manual.

> **When the HVCX1 option is used, the I/O lines are hardware configured as inputs or outputs. The software configuration must agree with the hardware configuration. The unit is protected from a configuration conflict, but the I/O lines will not respond properly.**
>
> **If you would like to experiment with the unit and the example programs, don't install the HVCX1 option until after experimenting with the unit to avoid software/hardware configuration conflicts.**

Once the board is set up, the HVCX1 board plugs into the DIN-96 connectors on the main Digital488/80A board labeled J1 and J4.

If the HVCX1 option was not factory installed, the DIN connectors have a jumper board plugged into them. Remove the jumper board.

Hold the HVCX1 board component side down, with the label FRONT toward the front of the Digital488/80A and the label REAR toward the back. Press the board into the DIN-96 connectors.

2.2.3 Mounting

Included with the Digital488/80A are accessories for rack or bench use.

2.2.3.1 Rack Mount

If rack mount installation is required, remove the two plastic screws from the predrilled holes on each side of the unit. Only remove the screws from the set of holes that will be toward the front of the unit (the unit can be mounted with the front or rear panel facing the front of the rack fixture).

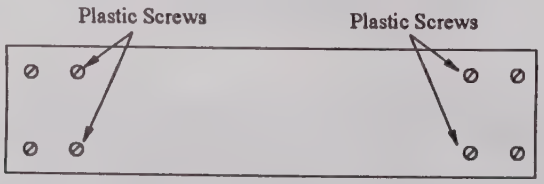


Figure 2.15: Enclosure Side View

Install the two rack ears using the enclosed screws.

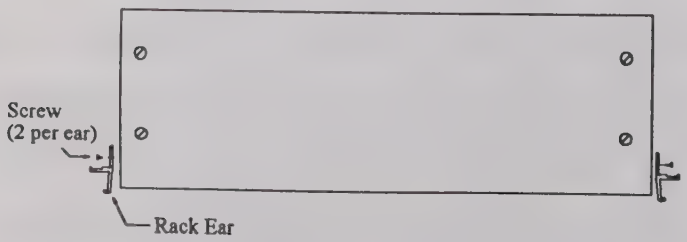


Figure 2.16: Rack Mounting Hardware (Top View)

2.2.3.2 Bench Top

If bench installation is required, the rubber feet can be installed on the underside of the unit approximately one inch from each corner.

2.2.4 Power Hookup; Power On

After plugging in the power cord, turn on the Digital488/80A by depressing the rear panel power switch. All of the front panel indicator lights should light for approximately one second while the Digital488/80A performs an internal ROM and RAM self check. At the end of this self check, all indicators should turn off except POWER.

If you obtain the above response, your Digital488/80A is alive and well. If all indicator lights remain on, a ROM error has occurred. If all of the indicator lights (except the power indicator) flash continuously, a RAM error has occurred. Try cycling the power to the Digital488/80A to determine that the error is repeatable.

If all of the indicator lights except POWER and ERROR go out, there was a checksum error on non-volatile RAM and the saved configurations may be lost. The checksum error condition may be cleared by doing a save of a configuration using the Save Configuration (Sn) command.

If the indicator lights do not flash and the POWER indicator does not light, make sure power is supplied to the unit. If the problem persists, refer to the Service Information section of this manual.

A self-test is initiated at power-on that checks for proper RAM and ROM operation.

2.3 IEEE 488 Bus Output Terminator Selection

The factory default setting for the Digital488/80A output terminating character is Carriage Return-Line Feed (CR LF). If necessary, it can be changed using the Terminator (Yn) command. To save the new setting as the power-up default, it must be saved with the Recall Configuration (S0X) command. See the Command Description section for more information on the Terminator and Recall Configuration commands. The only input terminating character accepted by the Digital488/80A is the Execute (X) command. All other terminators are ignored.

2.4 Front Panel Indicators

Six indicator lights on the front panel display the status of the Digital488/80A. The function of each indicator is described below.

TALK — On when the Digital488/80A is in the Talker state, off when the Digital488/80A is in the Idle or Listener state.

LISTEN — On when the Digital488/80A is in the Listener state, off when the Digital488/80A is in the Idle or Talker state.

SRQ — On when the Digital488/80A has generated a service request, off when no SRQ is pending (see Service Request Mask Command for more information).

ERROR On when an error has occurred, off when no error condition exists (see Error command for more information).

TEST Used in conjunction with the Test command to verify that communication has been established with the Digital488/80A.

POWER On when power is applied to the Digital488/80A and the power switch on the back panel is in the on position (depressed). Off if power is not present or the internal fuse or power supply is faulty.

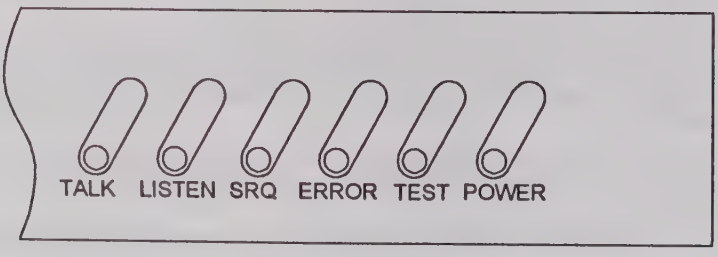


Figure 2.17: Front Panel Indicator Lights

2.5 Programming Examples

These examples use the IOtech Personal488 and the BASIC Keyboard Controller program listed in Appendix C. The Keyboard Controller program allows direct communication with the Digital488/80A (or any IEEE 488 device on the bus) and is the most convenient method of exercising the Digital488/80A and becoming familiar with the commands and their actions.

Verify that S1 is set for Dual Primary addressing mode and bus address 8 (the factory default). Connect an IEEE 488 cable (such as IOtech part number CA-7-3) from the Digital488/80A to the IEEE 488 connector on the Personal488. Turn on the Digital488/80A by depressing the rear panel power switch. All of the front panel indicator lights should be on for approximately one second while the Digital488/80A performs an internal ROM and RAM self-test. At the end of this self-test all indicators should turn off except power. Run the Keyboard Controller program and verify that the CMD prompt appears on the screen. The following is a listing of commands given to the Digital488/80A and the response received. Responses are separated by extra vertical space for clarity although these spaces do not actually appear when using the program.

This programming example will not work properly if the HVCX1 option is installed. Remove the HVCX1 board and replace it with the jumper board to run this example.

`CMD>`	Command prompt from Keyboard Controller program
`CMD>OUTPUT08;T1X`	Turn on the TEST indicator light
`CMD>OUTPUT08;T0X`	Turn off the TEST indicator light
`CMD>CLEAR`	Reset the Digital488/80A
`CMD>OUTPUT08;U0X`	Get status of Channel 0
`CMD>ENTER08`	
`1.0C0E0F0G0I000K0L0000M000P0R0Y0`	
	Status message shows all default options are in use
`CMD>OUTPUT09;U0X`	Get status of Channel 1
`CMD>ENTER09`	
`1.0C0E0F0G0I000K0M000P0R0Y0`	
	Status message shows all default options are in use
`CMD>OUTPUT08;C?`	Request Port Configuration Mode for Channel 0
`CMD>ENTER08`	
`C0`	C0 is returned, showing default port configuration
`CMD>OUTPUT08;C5X`	Configure all Channel 0 Ports as outputs
`CMD>OUTPUT08;C?`	Request Port Configuration Mode for Channel 0
`CMD>ENTER08`	
`C5`	C5 is returned, showing that the Cn command was executed
`CMD>OUTPUT09;C?`	Request Port Configuration Mode for Channel 1
`CMD>ENTER09`	
`C0`	C0 is returned because the default configuration of the ports on Channel 1 was not changed
`CMD>OUTPUT08;G2R0X`	Send output data when addressed to talk, channel 0
`CMD>OUTPUT08;D123ZX`	Send 123 to Channel 0
`CMD>ENTER08`	Read data from Channel 0
`0000000123`	
`CMD>OUTPUT08;A37X`	Set bit 37 on Channel 0
`CMD>ENTER08`	Read data from Channel 0
`1000000123`	Display shows bit 37 set and also the previous data

The above examples may also be repeated using Secondary addressing. To do this, turn off the Digital488/80A, set the rear panel switch (S1) for Secondary addressing mode and repeat the above commands, substituting 0800 for 08 and 0801 for 09 in all commands.

Additional Information

3.1 Handshaking/Control Lines

Six control lines on each channel enable handshaking of digital I/O data transfer to the Digital488/80A. They are automatically activated with the corresponding I/O activity and can also be independently activated with the Handshake (Hn) command. For all descriptions that follow, information on the function and action of a given line is true for both channels except where noted.

3.1.1 Clear (Pin 41)

The Clear output can be pulsed for approximately 50 microseconds after a Device Clear (DCL), Selected Device Clear (SDC), or Interface Clear (IFC) command has been sent on the bus (determined by the Format (Fn) mode of each channel). Execution of an SDC to either channel or a DCL affects both channels. The effect depends on whether either channel is in the High Speed Binary (F5) mode. If in the F5 mode, the channel(s) in the F5 mode have their command interpreter re-enabled and are returned to the ASCII Hexadecimal (F0) mode. All other parameters remain unchanged. In addition, the CLEAR lines on both channels are not pulsed. If neither channel is in the F5 mode, an SDC to either channel or a DCL returns both Channels to their power-up configuration and pulses both CLEAR lines for approximately 50 microseconds.

The Clear line is normally active high. The Invert command (I8) programs it active low. The Handshake command (H0) can pulse the Clear line of either channel, independent of any I/O operations.

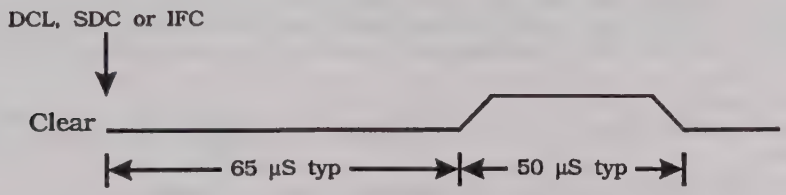


Figure 3.1: Timing Diagram for Clear Output

3.1.2 Data Strobe (Pin 42)

The Data Strobe output is pulsed for approximately 50 microseconds after new data are output on the I/O port. The Data Strobe line is normally active high, but may be programmed active low by the Invert command (I4). The Handshake command (H1) can pulse the Data Strobe line, independent of any I/O operations. The Data Strobe output is not pulsed when Bit Set (An) and Bit Clear (Bn) commands are executed.

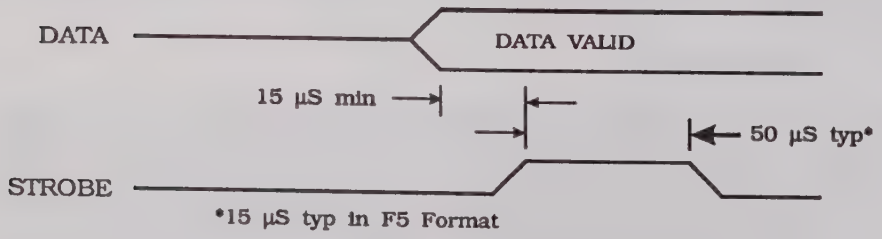


Figure 3.2: Timing Diagram for Strobe Output

3.1.3 External Data Ready [EDR] (46)

The External Data Ready [EDR] line is an edge sensitive input used to latch input data. It is used in conjunction with the Data Ready command (R1 or R2). The EDR signal must be at least 1 microsecond wide and must have a rise and fall time of less than one microsecond. The EDR line is normally rising-edge sensitive but can be programmed with the Invert command (I32) to be falling-edge sensitive. Refer to the Figure 3.3 for timing relationships.

When using the EDR line with the R1 command, data are not read when the Digital488/80A is addressed to talk as with R0. The Digital488/80A only outputs data when the EDR line transitions.

When using the EDR line with the R2 command, data are read and stored in an internal data buffer each time the EDR line transitions. The Digital488/80A only outputs data when there are data in the buffer to output.

EDR is not functional in the high speed binary (F5) format.

3.1.4 Inhibit (Pin 44)

The Inhibit output is asserted while data on the selected I/O port are being read into the I/O port buffer. This line is normally active high but may be programmed active low by the Invert command (I1). The Inhibit line can be programmed independent of any I/O operations with the Inhibit command (Qn). Refer to Figure 3.3 for timing relationships.

The Inhibit line is asserted once for each data read operation for all Format (Fn) modes except high speed binary (F5). In this mode, it is asserted for the first data read after the Digital488/80A is addressed to talk. On the last data byte transfer, the data are read again with Inhibit asserted in anticipation of another data transfer. If Inhibit is used to sequence external hardware, you should be aware that this line pulses N+1 times where N is the number of total (5 byte) data transfers.

The Inhibit Line is not pulsed when reading a particular bit status using the Status (Un) command.

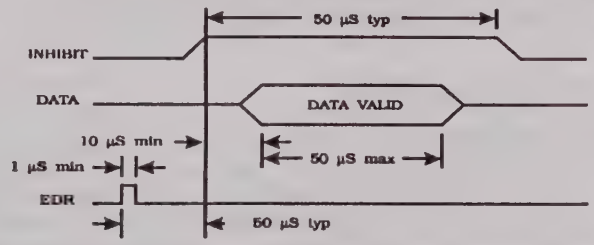


Figure 3.3: EDR Input & Inhibit Output Timing Diagram

3.1.5 Trigger (Pin 43)

The Trigger output is pulsed for approximately 50 microseconds after a GET (Group Execute Trigger) command is received from the bus controller. The trigger pulse is normally active high, but can be made active low with the Invert command (I2). The Handshake command (H2) can independently pulse the Trigger line, independent of any bus activity.

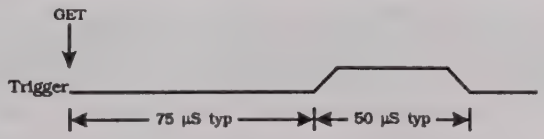


Figure 3.4: Timing Diagram for Trigger Output

3.1.6 Service (Pin 45)

The Service input is an edge sensitive input capable of generating a bus Service Request (SRQ). It is enabled with the Service Request Mask (M1) command and defaults to rising-edge sensitive. The Invert command (I64) can be used to program it to be falling-edge sensitive.

3.2 IEEE 488 Bus Implementation

The Digital488/80A implements many of the capabilities defined by the IEEE 488 1978 specification. These are discussed in the following sections. Those bus uniline and multiline commands that the Digital488/80A does not support or respond to include:

Remote Enable (REN)	Parallel Poll (PP)
Go to Local (GTL)	Parallel Poll Configure (PPC)
Local Lockout (LLO)	Parallel Poll Unconfigure (PPU)
Take Control (TCT)	Parallel Poll Disable (PPD)

3.2.1 My Talk Address (MTA)

In R0 mode, when the Digital488/80A is addressed to talk, it asserts Inhibit, reads the data from all ports, un-asserts Inhibit and outputs the data to the bus in the format as defined by the Format (Fn), Port (Pn) and Bus Input/Output (Gn) commands. The output bus terminators are appended to the output with the exception of the F4 and F5 formats. F4 does not append terminators. The output format of F5 is described separately. After output in the F0 through F4 formats, the Digital488/80A must be re-addressed to talk to perform subsequent reads.

In R1 mode, the Digital488/80A waits for an EDR transition on the selected channel before latching the data and formatting it for output. If the EDR line has changed state prior to being addressed to talk, the data read at the time of EDR is buffered for output when next addressed to talk. If EDR changes again before the previous EDR buffered data have been output, the Digital488/80A generates an EDR Overrun error and ignores the EDR read request. After output in the F0 through F4 formats, the Digital488/80A must be re-addressed to talk to perform subsequent buffered output of EDR captured data.

In R2 mode, the Digital488/80A waits for an EDR transition on the selected channel before latching the data and storing them in the internal data buffer. Up to 2000 readings may be latched and stored. The EDR and Inhibit lines function in the same manner as in the R1 mode. If an attempt is made to store more than 2000 readings, the Digital488/80A generates an Overrun error. After output in the F0 through F4 formats, the Digital488/80A/80 must be re-addressed to talk to perform subsequent buffered output of EDR captured data.

In any Rn mode, the Digital488/80A can request status using the Status (Un) command without affecting the data ports or the Inhibit line. After the requested status is output, the presently programmed Rn mode returns.

EDR cannot be used to capture data in the high speed binary format (F5). When addressed to talk in this format it asserts Inhibit, reads the data from all ports, un-asserts Inhibit and

outputs the binary data to the bus with EOI asserted on the fifth byte. When the last data byte is transferred, the data are read again in anticipation of another data transfer. If Inhibit is used to sequence external hardware, this line pulses N+1 times, where N is the number of total (5 byte) data transfers. In this format the Digital488/80A does not have to be re-addressed to talk to read the ports multiple times.

With all Fn formats, the data are output in a PORT5, PORT4, PORT3, PORT2, PORT1 sequence.

3.2.2 My Listen Address (MLA)

When the Digital488/80A is addressed to listen in the F0 through F4 format, it accepts characters from the active talker and interprets these characters as commands and command parameters. These commands are explained in Section 3.

In the high speed binary format (F5), the command interpreter is disabled. The Digital488/80A treats all bytes received as data to be output to the Digital I/O ports. Each time it receives five bytes or detects EOI it pulses the Data Strobe for approximately 15 microseconds. Data are expected in a PORT5, PORT4, PORT3, PORT2, PORT1 sequence.

If only two bytes are received, with EOI asserted on the second byte, the Digital488/80A updates PORT5 with the first byte received, PORT4 with the second and pulses the Data Strobe. Since the Digital488/80A treats all received characters as data, the Status (Un) command is not recognized.

3.2.3 Device Clear (DCL and SDC)

In the F0 thru F4 formats, Device Clear resets the Digital488/80A to power on defaults and pulses the Clear output line for approximately 50 microseconds.

If either Channel is in the high speed binary data format (F5), a DCL or SDC to either Channel enables the command interpreter on the Channel in the F5 mode and then changes the format to F0. All other parameters remain unchanged. In addition, the Clear output line is not pulsed by DCL or SDC when the either Channel is in the F5 mode. This is the only programmable method to exit the F5 format.

3.2.4 Group Execute Trigger (GET)

When the Digital488/80A recognizes a GET on either channel, it pulses the Trigger output lines on both channels for approximately 50 microseconds. If it is desired to pulse only one handshake line, the Handshake (H2) command should be issued to the desired channel.

3.2.5 Interface Clear (IFC)

IFC places the Digital488/80A in the Talker/Listener Idle State and pulses the Clear output line for approximately 50 microseconds.

3.2.6 Serial Poll Enable (SPE)

When Serial Poll Enabled, the Digital488/80A sets itself to respond to a serial poll with its serial poll status byte if addressed to talk. When the serial poll byte is accepted by the controller, any pending SRQs are cleared. The Digital488/80A continues to try to output its serial poll response until it is serial poll disabled by the controller.

3.2.7 Serial Poll Disable (SPD)

Disables the Digital488/80A from responding to serial polls by the controller.

3.2.8 Unlisten (UNL)

UNL places the Digital488/80A in the Listener Idle State.

3.2.9 Untalk (UNT)

UNT places the Digital488/80A in the Talker Idle State.

3.2.10 Serial Poll Response

Whenever the Digital488/80A generates an SRQ, a serial poll of either channel returns a serial poll status byte of at least 64 (decimal) showing that an SRQ has been generated by the Digital488/80A. In order to determine which channel generated the SRQ, use the Service Request Mask command (with the Query option M?) to recall what SRQ conditions are selected. The Digital488/80A returns M0 if SRQs are disabled or a response between either M1 through M7 or M16 through M23 depending on what SRQ conditions are selected. The following BASIC program illustrates one approach for handling service requests:

```
PRINT#1,"SPOLL08"                    Serial poll channel 0
INPUT#2,SRQ08
PRINT#1,"SPOLL09"                    Serial poll channel 1
INPUT#2,SRQ09
PRINT#1,"OUTPUT08;M?"                Get SRQ selection(s) from channel 0
PRINT#1,"ENTER08"
INPUT#2,M08$
PRINT#1,"OUTPUT09;M?"                Get SRQ selection(s) from channel 1
PRINT#1,"ENTER09"
INPUT#2,M09$
```

```
IF M08$="M0" THEN GOTO 500          SRQs disabled Channel 0, test Channel 1
IF M08$="M1" AND SRQ08 = 65 THEN GOTO 600
                                    Service input transition routine, Channel 0
IF M08$="M2" AND SRQ08 = 66 THEN GOTO 700
EDR                                 EDR input transition routine, Channel 0
IF M08$="M4" AND SRQ08 = 70 THEN GOTO 800
                                    Bus Error routine, Channel 0
```

Note: To allow the Digital488/80A to detect each of the above conditions, the Service Request Mask (Mn) command must be executed for each condition. More extensive comparisons of the Serial Poll Status Byte and the M? response are needed if multiple SRQ conditions are selected.

The unit may be programmed to generate an SRQ on power-on by selecting the M16 option (SRQ on Ready) of the Service Request Mask (Mn) command and saving this selection as part of the power-on configuration (configuration 0).

3.2.11 Data Transfer Speed Specifications

The Digital488/80A has been measured to perform at the speeds listed below. The testing was done in such a way that the limiting factor was the Digital488/80A and not the IEEE 488 bus controller driving it. All explanations discuss only one channel although the specifications given are valid for either channel.

3.2.12 Collection and Storage of Data

The Digital488/80A has an internal data buffer capable of storing 2000 readings of 40 bit data per channel. When operating in the R2 mode (data are latched and stored on an EDR transition), a transition on the External Data Ready input causes the Digital488/80A to collect the current value of all 40 bits and store the reading in the buffer. This collection and storage of 40 bits of data can occur at a maximum rate of once every 140 microseconds.

3.2.13 Output of Data to Ports

When sending data to the outputs on a channel of the Digital488/80A, the fastest Format mode is high-speed binary mode (F5). In this mode, five bytes of binary data are accepted from the bus and sent directly to the Digital488/80A outputs (all 40 bits). This constitutes one data transfer. The data transfer rate in this mode is approximately 1400 transfers per second.

3.2.14 IEEE 488 Bus Communication

The fastest data transfer rate across the IEEE bus to the IEEE 488 controller is approximately 1 byte every 140 microseconds whether reading data from the internal data buffer, or accepting commands or data from the bus.

3.2.15 Logic Levels - Base Unit

The data and handshake output lines drive two TTL loads. In addition, ports 1 and 2 on each Channel can function as 5 Volt CMOS-compatible outputs.

All input lines are less than 1.5 TTL loads. Normal precautions should be taken to limit the input voltages to –0.3 to +7.0 volts. All I/O lines are referenced to COMMON (Pin 50).

3.3 Digital Input/Output Ports

The Digital488/80A has 40 data lines per channel which can be programmed in groups of 8 bits as either input or output ports.

> **Note: When the HVCX1 option is used, the I/O lines are hardware configured as inputs or outputs. In this case, the software configuration must agree with the hardware configuration. The unit is protected from a configuration conflict, but the I/O lines will not respond properly.**

For convenience, you may want to program the Digital488/80A to power up with your configuration. Refer to the Configuration and Save Configuration commands. Input or output status is set using the Configuration (Cn) command. To save the input/output status as the new setting for the power-up default, it must be saved as part of the Recall Configuration (O0) command. To set the unit to a particular configuration upon power on, the Save Configuration (S0) command must be used once all the desired options have been selected.

3.3.1 Using the HVCX1 Option for High Voltage/High Current Outputs

Once the Channel port(s) are configured for high level operation, they can be used to drive devices such as switching relays, solenoids and displays. For example, a typical application may require a lamp and a relay to be driven by the Digital488/80A. The relay is connected to data line 28 (port 4) and the lamp is connected to data line 38 (port 5). These two ports must be configured as high level outputs. These devices would be connected as shown in Figure 3.5.

CAUTION

> **Never connect external high level devices to a Channel 1 port which has not first been configured for this purpose. Failure to do so may result in damage to the Digital488/80A.**

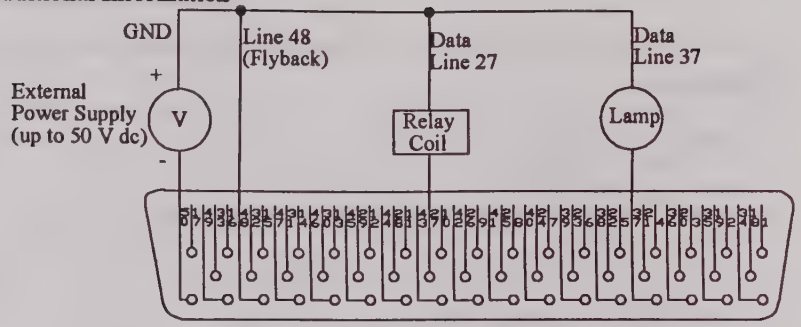


Figure 3.5: DB-50 Using High Level Outputs

CAUTION

Never connect external high level devices to any port on Channel 0. Failure to do so may result in damage to the Digital488/80A.

Since a relay is used for this application, the flyback terminal is connected to the positive terminal of the external power supply. Note also that the Ground lead of the power supply should be connected to pin 50 (GND) on the DB-50 connector. Remember that ports configured as high level outputs can only be used as outputs.

3.4 Digital I/O Port Pin Outs

Figure 3.6 illustrates one of the digital I/O DB-50 connectors as viewed from the rear of the Digital488/80A. Both connectors are identical.

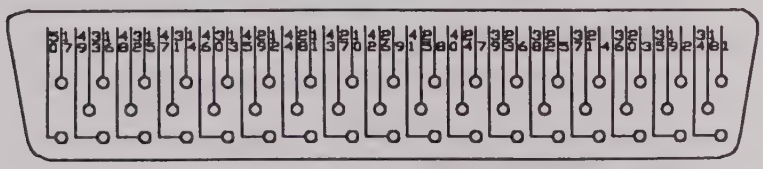


Figure 3.6: DB-50 Rear Panel Connector Pin-Outs

The HVCX1 option provides buffering for all the digital I/O lines on the Digital488/80A. This means that the HVCX1 option has 80 digital I/O lines that are divided into two 40-bit channels. Each channel is divided into five eight-bit ports. Six high level ASCII commands configure the ports to be all inputs, all outputs or combinations in between. Thus, a configuration can range from 80 input lines to 80 output lines or combinations in between in multiples of 8 bits.

Pin	Description
1 thru 8	DATA PORT1 (Input or Output). Pin 1 is bit 1 (LSB), Pin 8 is bit 8 (MSB). Least Significant Port
9 thru 16	DATA PORT2 (Input or Output). Pin 9 is bit 1 (LSB), Pin 16 is bit 8 (MSB).
17 thru 24	DATA PORT3 (Input or Output). Pin 17 is bit 1 (LSB), Pin 24 is bit 8 (MSB).
25 thru 32	DATA PORT4 (Input or Output). Pin 25 is bit 1 (LSB), Pin 32 is bit 8 (MSB).
33 thru 40	DATA PORT5 (Input or Output). Pin 33 is bit 1 (LSB), Pin 40 is bit 8 (MSB). Most Significant Port
41	CLEAR (Output).
42	DATA STROBE (Output).
43	TRIGGER (Output).
44	INHIBIT (Output).
45	SERVICE INPUT (Input).
46	EXTERNAL DATA READY [EDR] (Input).
47	Not Used
48	Used with HVCX1 only as Flyback (see text for more information).
49	+5 Volts (Do not exceed 50 mA load).
50	I/O COMMON (Gnd).

Table 3.1: Digital488/80A Pinouts

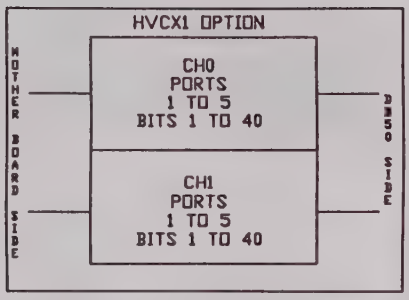


Figure 3.7: HVCX1 Option Block Diagram

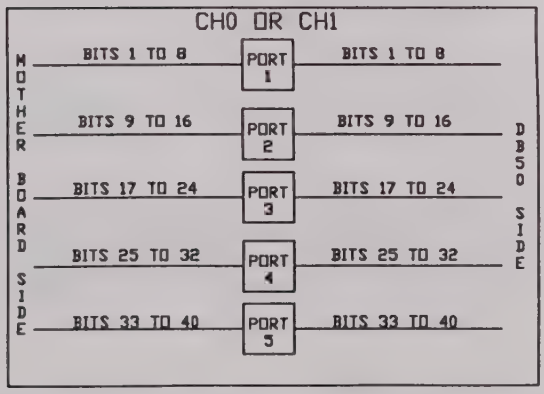


Figure 3.8: HVCX1 Channel Block Diagram (40 I/O Lines)

Note: The 2596 must be configured as an input or an output – not both.

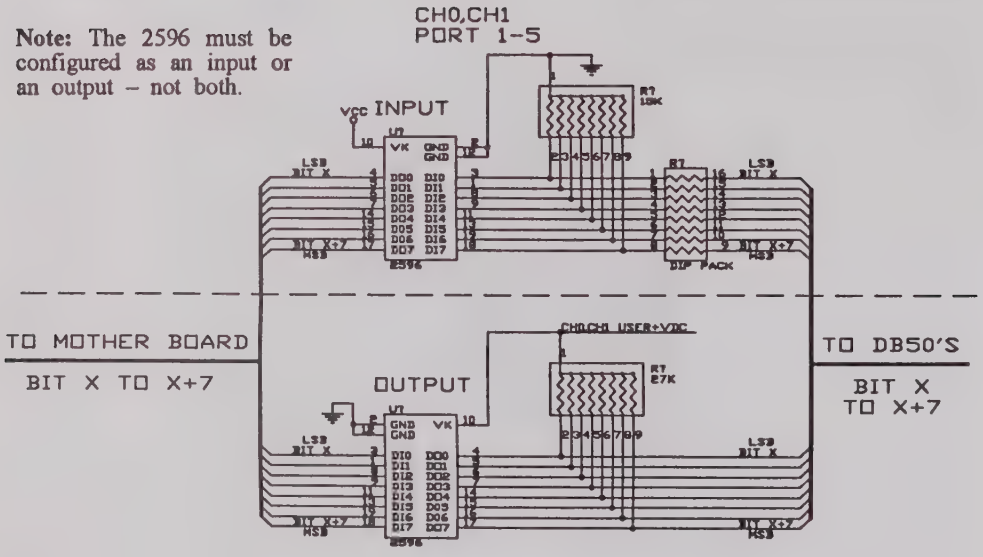


Figure 3.9: Typical HVCX1 Port Schematic

Command Descriptions

Control of the Digital488/80A is implemented with 23 bus commands, described here in detail. Examples are given for the commands using a personal computer (PC) functioning as an IEEE 488 controller by using the IOtech Personal488 PC/IEEE 488 board and associated driver software. All examples are given using GWBASIC or BASICA. The Digital488/80A bus address is set to 8 for all examples.

In order to establish communication with DRVR488 from BASIC, the following sequence must be used:

```
OPEN "\DEV\IEEEOUT" FOR OUTPUT AS #1
IOCTL#1,"BREAK"
PRINT#1,"RESET"
OPEN "\DEV\IEEEIN" FOR INPUT AS #2
```

All of the command examples that follow assume the driver has been properly opened and reset by the above sequence.

Most of the instructions offer a query (?) option. This option can be used to determine the present configuration or mode of a command previously executed. Any number of these query commands may be combined into one string to allow the user to construct a specialized status command requesting the Digital488/80A to return only the information of interest for a given application.

CAUTION

If you would like to experiment with the example programs, don't install the HVCX1 option until after experimenting with the unit to avoid software/hardware configuration conflicts. If the HVCX1 option is already installed, replace it with the jumper board to run the example programs.

Attention!

It is necessary that the EXECUTE command (X) follow all command strings sent to the Digital488/80A. No commands are executed until an X is received by the Digital488/80A. The only exceptions to this rule are the query commands. Any query commands do not have to be followed by an X.

Bit Set An

The Bit Set command programs a logic one output to a bit described by the argument n. Setting a bit may represent either a +5 volt or 0 volt output, depending on whether an Invert command (I16) has been sent. If data are active high (default condition), Bit Set outputs +5 volts. If multiple bits are to be set within the same command string, an Execute command (X) must be included after every Bit Set command.

The bit being set must have been configured as an output bit by the Configure command to be valid. The Strobe output line is not pulsed when the Bit Set command is sent.

An	Bit n (1 thru 40) is set to logic one
A?	Return n (1 thru 40) of last bit set

Examples:

This programming example will not work properly if the HVCX1 option is installed. Remove the HVCX1 board and replace it with the jumper board to run this example.

Using Dual Primary Addressing Mode to communicate with channel 0:

`PRINT#1,"CLEAR08"`	reset the Digital488/80A
`PRINT#1,"OUTPUT08;C5P0X"`	configure all ports as output, select all ports
`PRINT #1,"OUTPUT08;A22X"`	set bit 22 to a logic one
`PRINT#1,"OUTPUT08;A23XA24X"`	set bits 23 and 24 to a logic one

Using Secondary Addressing Mode to communicate with channel 0:

`PRINT#1,"CLEAR0800"`	reset the Digital488/80A
`PRINT#1,"OUTPUT0800;C5P0X"`	configure all ports as output, select all ports
`PRINT #1,"OUTPUT0800;A22X"`	set bit 22 to a logic one
`PRINT#1,"OUTPUT0800;A23XA24X"`	set bits 23 and 24 to a logic one

Bit Clear Bn

The Bit Clear command clears to a logic zero an output bit described by the argument n. Clearing a bit may represent either a 0 volt or +5 volt output, depending on whether an Invert command (I16) has been sent. If data are active high (default condition), Bit Clear outputs 0 volts. When multiple Bit Clear commands are used in the same command string, an Execute command (X) must follow each command.

The bit being cleared must have been defined as an output by the Configure (Cn) command in order to be valid. The Strobe output line is not pulsed when the Bit Clear command is sent.

Bn	Bit n (1 thru 40) is cleared to a logic 0
B?	Return n (1 thru 40) of last bit cleared

Example:

This programming example will not work properly if the HVCX1 option is installed. Remove the HVCX1 board and replace it with the jumper board to run this example.

`PRINT#1,"CLEAR08"`	reset the Digital488/80A
`PRINT#1,"OUTPUT08;C5P0X"`	configure all ports as output, select all ports
`PRINT#1,"OUTPUT08;A7XA8XA9X"`	set bits 7, 8, and 9 to +5 volts
`PRINT#1,"OUTPUT08;B7X"`	clear bit 7 to zero volts
`PRINT#1,"OUTPUT08;B8XB9X"`	clear bits 8 and 9 to zero volts

Configure Cn

Ports 1 through 5 are configured as inputs or outputs with the Configure command. Each port is eight bits wide. At power-on, all ports are initialized as inputs. The Configure command is usually the first command to be sent after power on. All ports programmed as outputs are set to a logic zero after receiving the Configure command. The actual output level is dependent on the Invert command (I16).

Cn Mode n (0 thru 5) defines which ports are input and output

Configure CommandCode	Port 5	Port 4	Port 3	Port 2	Port 1
C0	input	input	input	input	input
C1	input	input	input	input	output
C2	input	input	input	output	output
C3	input	input	output	output	output
C4	input	output	output	output	output
C5	output	output	output	output	output

C? Return present port configuration

Example:

This programming example will not work properly if the HVCX1 option is installed. Remove the HVCX1 board and replace it with the jumper board to run this example.

```
PRINT#1,"CLEAR08"                reset the Digital488/80A
PRINT#1,"OUTPUT08;C1X"           select port 1 as output, ports 2 thru 5 as
                                 inputs
PRINT#1,"OUTPUT08;C?"            request present port configuration
PRINT#1,"ENTER08",
INPUT#2,A$,
PRINT A$                         display shows C1
```

Data Dn...Z

The Data command outputs up to 40 bits of data to the output ports on a selected channel. The number of bits that can be sent with the Data command is limited by the number of bits programmed as outputs. For formats F0 through F3, if the amount of data sent is less than the number of bits programmed as outputs, the least-significant bits contain the data sent and the most-significant bits are cleared to logic zero. If a single port is selected with the Port command, only eight bits may be sent with the Data command. The Data Strobe output of the selected channel is pulsed for approximately 50 microseconds after new data are output on the selected ports of that channel.

For formats F0 through F3, data sent by the controller are contained within a prefix (D) and a suffix (Z). In format F4, the five bytes immediately following the prefix (D) are interpreted as data and the suffix (Z) is not used. For the high speed binary F5 format, all bytes received are treated as data and the prefix and suffix are not used. Refer to the Format (Fn) command for additional details.

Dn...Z n... represents the data to be outputted, terminated by Z (Note: in the F4 mode, the Z terminator is not allowed)

Example:

This programming example will not work properly if the HVCX1 option is installed. Remove the HVCX1 board and replace it with the jumper board to run this example.

```
PRINT#1,"CLEAR08"                    reset the Digital488/80A
PRINT#1,"OUTPUT08;C5P1G2R0X"         all ports are output, select port 1
PRINT#1,OUTPUT08;D55ZX"              send 55 to port 1
PRINT#1,"ENTER08",
INPUT#2,A$                           read data from port 1
PRINT A$                             display shows 55
PRINT#1,"OUTPUT08;P0X"               select all ports
PRINT#1,"OUTPUT08;D1234567890Zsend data to all 40 data lines
X"
PRINT#1,"ENTER08",
INPUT#2,A$                           read data from the Digital488/80A
PRINT A$                             display shows 1234567890
```

`PRINT#1,"CLEAR08"`	reset the Digital488/80A
`PRINT#1,OUTPUT08;C5P1G2R0X"`	all ports are output, select port 1
`PRINT#1,"OUTPUT08;D123ZX"`	send 12 bits of data to the least significant bits
`PRINT#1,"ENTER08",`	
`INPUT#2,A$`	read data from the Digital488/80A
`PRINT A$`	display shows `0000000123`
`PRINT#1,"OUTPUT08;P5D21ZX"`	select port 5 only
`PRINT#1,"OUTPUT08;P0X"`	select all ports
`PRINT#1,"ENTER08",`	
`INPUT#2,A$`	read data from the Digital488/80A
`PRINT A$`	display shows `2100000123`

Error E?

The Error command is used to determine the present error condition on the selected channel of the Digital488/80A. After execution of the Error command, the present error condition is cleared. The error condition is also cleared by executing the Status (U0) command. When an error has occurred, the Error light on the front panel of the Digital488/80A turns on. The Error light stays on until an Error (E?) or Status (U0) command is executed.

If a Non-Volatile RAM Checksum Failure (E5) occurs, it can only be cleared by doing a save of a configuration using the Save Configuration (Sn) command. Executing this command updates the checksum to a known value and clears the E5 error condition.

E?	Return and clear present error condition

After execution of the Error command, the Digital488/80A returns one of the following error codes:

E0	No Error has occurred
E1	Unrecognized command (W4X, J5, etc.)
E2	Invalid Command Parameter (A62X, C10X, etc.)
E3	Conflict Error (outputting data to an input port, sending excess data)
E4	Not used
E5	Non-Volatile RAM Checksum Failure
E6	Internal Data Buffer Overrun (attempting to store data in a full buffer)

Examples:

`PRINT#1,"CLEAR08"`	reset the Digital488/80A
`PRINT#1,"OUTPUT08;E?"`	request error status
`PRINT#1,"ENTER08",`	
`INPUT#2,A$,`	
`PRINT A$`	display shows E0, no errors have occurred
`PRINT#1,"OUTPUT08;W5X"`	send an illegal command to the Digital488/80A
`PRINT#1,"OUTPUT08;E?"`	request error status (error light should be on)
`PRINT#1,"ENTER08",`	
`INPUT#2,A$,`	
`PRINT A$`	display shows E1-Unrecognized Command (error light should be off)

`PRINT#1,"OUTPUT08;E?"`	request error status
`PRINT#1,"ENTER08",`	
`INPUT#2,A$,`	
`PRINT A$`	display shows E0 because error condition has been cleared after reading error status
`PRINT#1,"OUTPUT08;C8X"`	send a command with an invalid option
`PRINT#1,"OUTPUT08;E?"`	request error status (error light should be on)
`PRINT#1,"ENTER08",`	
`INPUT#2,A$,`	
`PRINT A$`	display shows `E2-Invalid Parameter` (error light should be off)
`PRINT#1,"OUTPUT08;G0C5P1X"`	configure all ports as outputs, select port 1
`PRINT#1,"OUTPUT08;D123456ZX"`	send too much data for one port
`PRINT#1,"OUTPUT08;E?"`	request error status (error light should be on)
`PRINT#1,"ENTER08",`	
`INPUT#2,A$,`	
`PRINT A$`	display shows `E3-Conflict Error` (error light should be off)

Format Fn

The Format command determines the method by which input and output data are described. Six data formats are available.

F0	ASCII Hexadecimal (4 bits per character)
F1	ASCII Character (4 bits per character)
F2	ASCII Binary (1 bit per character)
F3	ASCII Decimal (8 bits per number)
F4	Binary (each byte represents 8 bits)
F5	High Speed Binary (each byte represents 8 bits)
F?	Return present data format

F0 Format - ASCII Hexadecimal

In the default F0 format, the data are described in ASCII hexadecimal, with each character having a value from 0 thru 9 or A thru F. Each ASCII character describes 4 bits of data.

F0 Character	Decimal Equivalent	F0 Character	Decimal Equivalent
0	0	8	8
1	1	9	9
2	2	A	10
3	3	B	11
4	4	C	12
5	5	D	13
6	6	E	14
7	7	F	15

Data received for output to the digital ports must be contained within a prefix (D) and a suffix (Z). If the amount of data sent is less than the number of bits programmed as outputs, the least-significant bits contain the data sent and the most-significant bits are cleared to logic zero. If the data sent are greater than the number of bits programmed for output or selected by the Port (Pn) command, the Digital488 generates a conflict error (E3) and ignores the entire command string up to the next X. The Data Strobe output is pulsed for approximately 50 microseconds after new data are output on the selected port(s).

When the Digital488/80A is addressed to talk (in Data Ready mode R0), it asserts Inhibit, reads the data from all ports, unasserts Inhibit and outputs the number of characters determined by the Bus Input/Output (Gn) and Port (Pn) commands. Leading

zeros are not suppressed and the bus terminators are appended to the output. After output the Digital488/80A must be re-addressed to talk to perform subsequent reads. EDR (in R1 or R2 mode) may also be used to capture data in this format.

Example:

This programming example will not work properly if the HVCX1 option is installed. Remove the HVCX1 board and replace it with the jumper board to run this example.

`PRINT#1,"CLEAR08"`	reset the Digital488/80A
`PRINT#1,"OUTPUT08;R0P0C2G2X"`	configure ports 1 and 2 as output
`PRINT#1,"OUTPUT08;F0X"`	select ASCII Hexadecimal format
`PRINT#1,"OUTPUT08;D4E6BZX"`	output hexadecimal 4E6B to ports 1 and 2
`PRINT#1,"ENTER08",`	
`INPUT#2,A$`	read data from the Digital488/80A
`PRINT A$`	display shows 4E6B

F1 Format - ASCII Character

In the F1 format, the data are coded and transmitted in ASCII Characters with the four least significant bits of each ASCII character representing four bits of data.

F1 Character	Decimal Equivalent	F1 Character	Decimal Equivalent
0	0	8	8
1	1	9	9
2	2	:	10
3	3	;	11
4	4	<	12
5	5	=	13
6	6	>	14
7	7	?	15

Data received for output to the digital ports must be contained within a prefix (D) and a suffix (Z). If the amount of data sent is less than the number of bits programmed as outputs, the least significant bits contain the data sent and the most

significant bits are cleared to logic zero. If the data sent are greater than the number of bits programmed for output or selected by the Port (Pn) command, the Digital488/80A generates a conflict error (E3) and ignores the entire command string up to the first X. The Data Strobe output is pulsed for approximately 50 microseconds after new data is output on the selected port(s).

When the Digital488/80A is addressed to talk (in R0 mode), it asserts Inhibit, reads the data from all ports, unasserts Inhibit and outputs the number of characters determined by the Gn and Pn commands. Leading zeros are not suppressed and the bus terminators are appended to the output. After output the Digital488/80A must be re-addressed to talk to perform subsequent reads. EDR (in R1 or R2 mode) may also be used to capture data in this format.

Example:

This programming example will not work properly if the HVCX1 option is installed. Remove the HVCX1 board and replace it with the jumper board to run this example.

```
PRINT#1,"CLEAR08"                reset the Digital488/80A
PRINT#1,"OUTPUT08;C5G2R0P0X"     all ports are outputs, select all ports
PRINT#1,"OUTPUT08;F1X"           select ASCII Character format
PRINT#1,"OUTPUT08;D4>6ZX"        send 4>6 to the Digital488/80A
PRINT#1,"ENTER08",
INPUT#2,A$                       read data from the Digital488/80A
PRINT A$                         display shows 00000004>6
PRINT#1,"OUTPUT08;D1??2ZX"       send 1??2 to the Digital488/80A
PRINT#1,"ENTER08",
INPUT#2,A$                       read data from the Digital488/80A
PRINT A$                         display shows 0000001??2
```

F2 Format - ASCII Binary

In the F2 format, each data bit is described with an ASCII 0 or 1. Each byte is formatted in two 4-bit multiples separated by semicolons.

F2 String	Decimal Equiv	F2 String	Decimal Equiv
0000;0000	0	0000;1001	9
0000;0001	1	0000;1010	10
0000;0010	2	0000;1011	11
0000;0011	3	0000;1100	12
0000;0100	4	0000;1101	13
0000;0101	5	0000;1110	14
0000;0110	6	0000;1111	15
0000;0111	7	1000;0001	129
0000;1000	8	1111;1111	255

Data received for output to the digital ports must be contained within a prefix (D) and a suffix (Z) and each 4-bit quantity must be separated by semicolons. Leading zeros are not required. If the amount of data sent is less than the number of bits programmed as outputs, the least-significant bits contain the data sent and the most-significant bits are cleared to logic zero. If the data sent are greater than the number of bits programmed for output or selected by the Pn command, the Digital488/80A generates a conflict error (E3) and ignore the entire command string up to the next X. The Data Strobe output is pulsed for approximately 50 microseconds after new data are output on the selected port(s).

When the Digital488/80A is addressed to talk (in R0 mode) it asserts Inhibit, reads the data from all ports, unasserts Inhibit and outputs the number of characters determined by the Gn and Pn commands. Leading zeros are not suppressed and the bus terminators are appended to the output. After output the Digital488/80A must be re-addressed to talk to perform subsequent reads. EDR (in R1 or R2 mode) may also be used to capture data in this format.

Example:

This programming example will not work properly if the HVCX1 option is installed. Remove the HVCX1 board and replace it with the jumper board to run this example.

```
PRINT#1,"CLEAR08"                    reset the Digital488/80A
PRINT#1,"OUTPUT08;C5G2R0P1X"         all ports are outputs, select port 1
```

```
PRINT#1,"OUTPUT08;F2X"          select ASCII/binary mode
PRINT#1,"OUTPUT08;D1;1011ZX"    send binary data
PRINT#1,"ENTER08",
INPUT#2,A$                      read data from the Digital488/80A
PRINT A$                        display shows 0001;1011
```

F3 Format - ASCII Decimal

In the F3 format, the data are described in decimal 8 bit multiples and transmitted in ASCII. Each decimal number (0 to 255) to be output must be separated by semicolons.

F3 Number	Decimal Equivalent	F3 Number	Decimal Equivalent
000	0	008	8
001	1	009	9
002	2	010	10
003	3	020	20
004	4	100	100
005	5	200	200
006	6	210	210
007	7	255	255

Data received for output to the digital ports must be contained within a prefix (D) and a suffix (Z). If the amount of data sent is less than the number of bits programmed as outputs, the least-significant bits contain the data sent and the most-significant bits are cleared to logic zero. If the data sent are greater than the number of bits programmed for output or selected by the Pn command, the Digital488/80A generates a conflict error (E3) and ignores the entire command string up to the next X. The Data Strobe output is pulsed for approximately 50 microseconds after new data are output on the selected port(s).

When the Digital488/80A is addressed to talk (in R0 mode) it asserts Inhibit, reads the data from all ports, unasserts Inhibit and outputs the number of characters determined by the Gn and Pn commands. Leading zeros are not suppressed and the bus terminators are appended to the output. After output the Digital488/80A must be re-addressed to talk to perform subsequent reads. EDR (in R1 or R2 mode) may also be used to capture data in this format.

Example:

This programming example will not work properly if the HVCX1 option is installed. Remove the HVCX1 board and replace it with the jumper board to run this example.

```
PRINT#1,"CLEAR08"               reset the Digital488/80A
```

```
PRINT#1,"OUTPUT08;F3X"            select decimal mode
PRINT#1,"OUTPUT08;C5G2R0P0X"      all ports are outputs, select all ports
PRINT#1,"OUTPUT08;D240;165ZX"     send data to the Digital488/80A
PRINT#1,"ENTER08"                 read data from the Digital488/80A
INPUT#2,A$,
PRINT A$                          display shows 000;000;000;240;165
```

F4 Format - Binary

In the F4 binary format, no error checking is performed and caution must be exercised when using this mode to avoid locking the IEEE bus.

When addressed to listen, the Digital488/80A expects the D prefix followed by five bytes of data beginning with PORT5 without the Z suffix. If any digital I/O port is configured as an input, the data to that input port are ignored.

When the Digital488/80A is addressed to talk (in R0 mode), it asserts Inhibit, reads the data from all ports, unasserts Inhibit and outputs five bytes beginning with PORT5 with EOI asserted on the last bye. Bus terminators, with the exception of EOI, are not appended to the output. After output the Digital488/80A must be re-addressed to talk to perform subsequent reads. EDR (in R1 or R2 mode) may also be used to capture data in this format.

F5 Format - High Speed Binary

In the F5 high speed binary format, the command interpreter is disabled. When addressed to listen, the Digital488/80A treats all bytes received as data to be output to the Digital I/O ports. Each time it receives five bytes or detects EOI asserted, it pulses the Data Strobe for approximately 15 microseconds. Data are expected in a PORT5, PORT4, PORT3, PORT2, PORT1 sequence. If only two bytes are received, with EOI asserted on the second byte, the Digital488/80A updates PORT5 with the first byte received, PORT4 with the second and pulses the Data Strobe. Since all received characters are treated as data, no commands are recognized.

To place the Digital488/80A in the F5 format, the three character string F5X should be the last command sent to the interface without terminators. Any characters appended to this command, such as carriage return or line feed, are considered data and the output ports reflects those character values (see the following program example).

When addressed to talk in this format it asserts Inhibit, reads the data from all ports, unasserts Inhibit and outputs the binary data to the bus with EOI asserted on the fifth byte. When the last data byte is transferred, the data are read again in anticipation of another data transfer. If Inhibit is used to sequence external hardware, this line pulses N+1 times, where N is the number of total (5 byte) data transfers. In this format the Digital488/80A does not have to be re-addressed to talk to read the ports multiple times. EDR cannot be used to capture data in the F5 high speed binary format.

The only programmable method to exit the F5 high speed binary format is device clear (DCL) or Selected Device Clear (SDC). When received, it enables the command interpreter and changes the format to F0. All other parameters remain unchanged.

In addition, the Clear output line is not pulsed by DCL or SDC when the Digital488/80A is in F5.

When either channel (or both) is in the F5 mode, a DCL or an SDC to either channel clears the F5 mode. It does not reset the interface. It only takes the unit out of the F5 mode and places it in the F0 mode.

Example:

```
PRINT#1,"OUTPUT08#3F5X"          put Channel 0 in F5 mode
PRINT#1,CHR$(255),CHR$(125)      send data
```

Note: The #3 tells DRVR488 to send only the three bytes F5X without terminators. If this was not done, the first data bytes to appear at the output of the Digital488/80A would be CR LF or whatever terminators were selected.

Bus Input/Output Gn

The Bus Output command determines whether input port data, output port data, both input and output port data, or buffered data are transmitted on the bus when the Digital488/80A is addressed to talk. The amount of data sent is dependent on the Pn command.

The G0 default mode causes all input and output port data to be sent to the controller when addressed to talk. The G1 mode causes only data from the ports programmed as inputs to be returned when addressed to talk. The G2 mode causes only data from ports programmed as outputs to be returned when addressed to talk.

The G3 mode causes data in the data buffer to be returned when addressed to talk. Both input and output data (all 40 bits) are returned in this mode. If the count of readings in the buffer is 0, the Digital488/80A holds off the bus until the next transition of EDR and another reading has been collected and buffered. After the EDR transition, it returns that newly collected reading. The Digital488/80A must be readdressed to perform subsequent buffered output of EDR captured data. The G4 mode is identical to the G3 format except that the Digital488/80A performs subsequent buffered output of EDR captured data until the controller stops handshaking.

If all ports are programmed as outputs with G1 selected and the Digital488/80A is addressed to talk, nothing is transmitted and the bus hangs. The converse also causes the bus to hang with all ports programmed as inputs and G2 selected.

G0	Input and output port data are sent when addressed to talk
G1	Only input port data are sent when addressed to talk
G2	Only output port data are sent when addressed to talk
G3	Buffered input and output data are sent when addressed to talk
G4	Continuous buffered input and output data are sent when addressed to talk
G?	Return present Bus Input/Output mode

Example:

This programming example will not work properly if the HVCX1 option is installed. Remove the HVCX1 board and replace it with the jumper board to run this example.

```
PRINT#1,"CLEAR08"                reset the Digital488/80A
PRINT#1,"OUTPUT08;P0C1R0X"       port 1 as output, ports 2-5 as input
PRINT#1,"OUTPUT08;G1X"           select only input ports for data
PRINT#1,"ENTER08",
INPUT#2,A$                       read data from the input ports
PRINT A$                         display shows FFFFFFFF (data are
                                 dependent on what is connected).
```

Handshake Hn

The Handshake control command enables software control of the handshake lines, independent of any other I/O operations. When the Digital488/80A receives an Hn command, the respective handshake line is pulsed for approximately 50 microseconds. It returns to its steady-state condition after pulsing. The Invert (In) command may be used to change the active state of any of the handshake lines.

H0	The Clear line is pulsed
H1	The Strobe line is pulsed
H2	The Trigger line is pulsed
H?	Return last Handshake command executed

Example:

```
PRINT#1,"OUTPUT08;H1X"      pulse the Strobe line on Channel 0
```

Invert In

The Invert command is used to change the polarity of the handshake and data lines. At power up all handshake and control lines are active high (logic one = + 5 volts). The Invert command can selectively change the polarity of each of the handshake lines, and of the data lines. If multiple Invert commands are contained within the same string, an Execute command (X) should be included between each Invert command. An alternative is to add the values of each Invert command desired, and send one command with the sum of the desired commands. The Invert commands are ORed together as received. To delete any one command, it is necessary to program the default mode I0, then reprogram the desired commands.

I0	All control lines are active high, all data lines are high true
I1	Inhibit output is active low
I2	Trigger output is active low
I4	Data Strobe output is active low
I8	Clear output is active low
I16	Data are low true
I32	EDR input is falling-edge sensitive
I64	Service input is falling-edge sensitive
I?	Return last Invert command executed

Example:

`PRINT#1,"CLEAR08"`	reset the Digital488/80A
`PRINT#1,"OUTPUT08;I32XI64X"`	select EDR and Service input as falling-edge sensitive

Note:

`PRINT#1,"OUTPUT08;I96X"`	performs the same function as above

End or Identify (EOI) Kn

The EOI line is one of five interface management lines on the IEEE 488 bus. It is used by a talker to indicate the end of a multiple byte transfer sequence. The controller can change the EOI mode by programming the Digital488/80A from the bus. In the K0 mode, the EOI line is asserted by the Digital488/80A on the last byte of every bus output string. In the K1 mode the EOI function is disabled (except when using the binary modes F4 and F5).

K0	EOI enabled, assert EOI on last byte transferred
K1	EOI disabled, do not assert EOI on last byte transferred
K?	Return present EOI selection

Example:

```
PRINT#1,"OUTPUT08;K1X"      disables EOI on last byte
PRINT#1,"OUTPUT08;K?"       request EOI selection from Digital488/80A
PRINT#1,"ENTER08",
INPUT#2,A$,
PRINT A$                    display shows K1
```

Buffer Ln

The Buffer command is used to clear the internal data buffer or to determine the number of readings in the buffer. It is used primarily with the Buffered Data Ready command (R2), although it may be used at any time to determine the buffer status.

L0	Clear the internal data buffer
L?	Return number of port readings in the internal data buffer

Examples:

```
PRINT#1,"OUTPUT08;C0P0R2G3X"        buffer data to be sent when addressed to talk

                                    collect some data
PRINT#1,"OUTPUT08;L?X"              request buffer status of Channel 0
PRINT#1,"ENTER08",
INPUT#2,COUNT$,
COUNT$ = RIGHT$(COUNT$,4),
COUNT = VAL(COUNT$),
PRINT COUNT                         display shows 24 (24 port readings in buffer)

DIM B(100)                          store readings in an array
FOR I = 1 TO COUNT                  readings are stored as integers
PRINT#1,"ENTER08",
INPUT#2,A$,
B(I) = VAL(RIGHT$(A$,4)),
NEXT I,
PRINT#1,"OUTPUT08;L?X"              request buffer status
PRINT#1,"ENTER08",
INPUT#2,COUNT$,
PRINT COUNT$                        display shows L000 since all data have been read
```

Service Request Mask (SRQ) Mn

The Service Request (SRQ) Mask command is used by the Digital488/80A to alert the controller to one of several conditions described below. Multiple Service Request conditions can be enabled simultaneously by issuing them separately or by combining them in one command string. If multiple Mn commands are contained within the same command string, each command should be followed by an Execute (X) command. The programmed SRQ modes remain enabled until the M0 (SRQ disabled) command is sent, or the controller sends a Device Clear (DCL) or Selected Device Clear (SDC).

M0	SRQ is disabled
M1	SRQ on Service input transition
M2	SRQ on EDR input transition
M4	SRQ on bus error
M16	SRQ on Ready

M0 default mode disables the SRQ function, preventing the Digital488/80A from generating a Service Request.

M1 enables the Digital488/80A to generate a Service Request when the Service Input line makes a transition. Refer to the Invert (I64) command description for programming the polarity of the Service input line.

M2 enables the Digital488/80A to generate a Service Request when the EDR input makes a transition. Refer to the Invert (I32) command description for programming the polarity of the EDR input line.

M4 enables the Digital488/80A to generate a Service Request when a bus error occurs. The most common bus error is sending an invalid command to the Digital488/80A. For example, selecting F6 format causes the Digital488/80A to generate a Service Request when the M4 mode is selected.

M16 enables the Digital488/80A to generate a Service Request when it has completed the execution of a set of commands from the bus controller. This may be used by the controller to assure the completion of a set of commands before sending a subsequent set of commands.

Like the Invert (In) commands, the Service Request Mask commands are also ORed together as received. To delete any one command, it is necessary to program the default mode M0, then reprogram the desired commands.

Example:

`PRINT#1,"CLEAR08"`	reset the Digital488/80A
`PRINT#1,"OUTPUT08;M4X"`	select SRQ on Bus error
`PRINT#1,"OUTPUT08;F7X"`	send an invalid command
	Note: ERROR and SRQ indicator lights should illuminate
`PRINT#1,"CLEAR08"`	reset the Digital488/80A
`PRINT#1,"OUTPUT08;M1XM4X"`	select SRQ on Bus Error and SRQ on Service Input Transition
`PRINT#1,"OUTPUT08;M5X"`	This has the same effect as the command above where `M1X` plus `M4X` equals `M5X`

Recall Configuration On

The Recall Configuration command is used to load a previously saved configuration from non-volatile RAM and use it as the present configuration. The argument of the command (n) may range from 0 to 100. As mentioned under the Save Configuration (Sn) command, configuration 0 is the power-on configuration. Any load of a configuration not previously saved with the Save Configuration command configures the channel with the default setting for each command (i.e. Format F0, Terminator Y0 etc.) Issuing an O? command returns the configuration number last recalled.

On	Load configuration n for use as present channel configuration
O?	Return present channel configuration number (n)

Example:

`PRINT#1,"CLEAR08"`	reset the Digital488/80A
`PRINT#1,"OUTPUT08;O5X"`	load configuration 5 from non-volatile RAM for use as present configuration

Port Pn

The Port command determines which port is selected for data input/output. In the default mode (P0), all ports are selected. The P1 thru P5 commands select a specific eight bit port.

It is recommended that the Bus Input/Output (Gn) command be used with the P0 mode to select which type of port data (input or output) is sent when the Digital488/80A is addressed to talk. Data in modes P1 through P5 are input or output in groups of eight bits.

P0	All five ports are selected
P1	Port 1 is selected
P2	Port 2 is selected
P3	Port 3 is selected
P4	Port 4 is selected
P5	Port 5 is selected
P?	Return present port selection

Example:

```
PRINT#1,"CLEAR08"          reset the Digital488/80A
PRINT#1,"OUTPUT08;P4X"     select port 4
```

Inhibit Qn

The Inhibit control command allows software control of the Inhibit line, independent of any other I/O activities. The 'set' and 'clear' levels of the Inhibit line are determined by the Invert (`In`) command.

`Q0`	Clear the Inhibit line (return to unasserted state)
`Q1`	Set the Inhibit line (place in the asserted state)
`Q?`	Return the last Inhibit command executed

Example:

`PRINT#1,"CLEAR08"`	reset the Digital488/80A
`PRINT#1,"OUTPUT08;Q1X"`	set the Inhibit line on Channel 0

Data Ready Rn

The Data Ready command enables digital input port data to be latched. When used in conjunction with the Service Request Mask (M2) command, the External Data Ready line can both latch the input data and generate an SRQ to signal the controller that new data are available.

In the default mode (R0), data are read when the Digital488/80A is addressed to talk. In the R1 mode, it waits for an External Data Ready (EDR) transition on the selected channel before latching the data and formatting the data for output. If the Digital488/80A is addressed to talk before EDR is asserted, the bus hangs up until the EDR pulse occurs. Once EDR is asserted the data remain latched until the interface is addressed to talk and the data are read by the controller. If EDR transitions again before the previous EDR buffered data have been read, the Digital488/80A generates a Buffer Overrun error (E6) and ignores the EDR read request. The Inhibit line is asserted once for each data read operation for all format (Fn) modes except high speed binary (F5).

In the R2 mode, successive readings (up to 2000) may be latched and stored in the internal data buffer. The number of readings in the buffer may be found using the L? command. The EDR and Inhibit lines function in the same manner as in the R1 mode. If an attempt is made to store more than 2000 readings in the internal buffer, the Digital488/80A generates a Buffer Overrun error (E6).

The EDR signal must be at least 1 microsecond wide and should have a rise and fall time of less than 1.0 microsecond. The EDR line defaults to rising-edge sensitive, but can be changed to falling-edge sensitive with the Invert command (I32). EDR cannot be used to capture data in the F5 high speed binary format.

R0	Data are not latched, and are read whenever the Digital488/80A is addressed to talk
R1	Data are latched on an EDR transition
R2	Data are latched and buffered on an EDR transition
R?	Return present programmed Data Ready mode

Example:

This programming example will not work properly if the HVCX1 option is installed. Remove the HVCX1 board and replace it with the jumper board to run this example.

`PRINT#1,"CLEAR08"`	reset the Digital488/80A
`PRINT#1,"OUTPUT08;R1X"`	data are read only after a rising-edge signal is applied to the EDR line (only one reading is stored in buffer and must be read before another reading can be taken)
`PRINT#1,"CLEAR08"`	reset the Digital488/80A
`DIM B$(100)`	dimension an array to store buffered data
`PRINT#1,"OUTPUT08;C0P0X"`	all ports are inputs, select all ports
`PRINT#1,"OUTPUT08;R2X"`	data are latched and buffered on EDR transition
`PRINT#1,"OUTPUT08;G3X"`	buffered I/O data sent when addressed to talk
	50 readings (one reading per EDR transition) are taken and stored in the internal buffer
`FOR I = 1 TO 50`	read the internal buffer and store the data in the array
`PRINT#1,"ENTER08",`	
`INPUT#2,B$(I),`	
`NEXT I,`	

Save Configuration Sn

The Save Configuration command is used to save a non-volatile copy of the present channel configuration for later recall with the Recall Configuration (On) command. The argument (n) for this command may range from 0 to 100, allowing 101 different configurations to be saved in battery-backed-up RAM. Upon power on, the default configuration is the S0 configuration. The present state of all output bits is read before saving. When the configuration is restored using the Recall Configuration (On) command, the outputs are set to the same values they had at the time the Sn command was executed. If it is desired that the outputs be set to a specific state as part of a configuration, the Data (Dn...Z) command should be used just prior to the Save Configuration command to insure that the data bits are set to the proper state upon retrieval of the configuration.

Sn	Save present configuration as configuration n (n = 0 to 100)
S?	Return the last configuration number (n) saved (not the present configuration loaded)

Example:

This programming example will not work properly if the HVCX1 option is installed. Remove the HVCX1 board and replace it with the jumper board to run this example.

`PRINT#1,"CLEAR08"`	reset the Digital488/80A
`PRINT#1,"OUTPUT08;C5X"`	configure all ports as outputs
`PRINT#1,"OUTPUT08;F3X"`	select ASCII Decimal data format
`PRINT#1,"OUTPUT08;P4X"`	select port 4
`PRINT#1,"OUTPUT08;S57X"`	save present channel configuration as configuration 57

Test Tn

The Test command is used to verify that communication has been established with the Digital488/80A. If the T1 command is executed, the front panel TEST indicator should light. If the T0 command is then executed, the TEST indicator light should turn off. This command can be used to test the connection between the IEEE Controller and the Digital488/80A.

T1	Turn on Test indicator light
T0	Turn off Test indicator light

Example:

PRINT#1,"CLEAR08"	reset the Digital488/80A
PRINT#1,"OUTPUT08;T1X"	turn on the TEST indicator light. TEST indicator light should illuminate
PRINT#1,"OUTPUT08;T0X"	turn off the TEST indicator light. TEST indicator light should extinguish

Status Un

The Status (U0) command causes the Digital488/80A to send the status message when next addressed to talk. The status of the Digital488/80A may be read at any time without interfering with normal operation. Any error conditions are cleared after the status string is read by the controller. The Status (Un) command enables the controller to read any single bit from the I/O ports (n = 1 through 40).

U0	Send the Digital488/80A status when next addressed to talk
Un	Send the status of bit n (n = 1 through 40) when next addressed to talk
U?	Return last status command executed

The format of the status byte returned by the Digital488/80A after receiving a U0 command is as follows:

```
*.*C#E#F#G#I###K#L####M###P#R#Y#
```

where each # equals the number corresponding to that command. The leading information *.* is the revision level of the Digital488/80A firmware.

Example:

```
PRINT#1,"CLEAR08"          reset the Digital488/80A
PRINT#1,"OUTPUT08;U0X"     request the status of the Digital488/80A
PRINT#1,"ENTER08",
INPUT#2,A$                 read the status byte
PRINT A$                   display =
                           1.0C0E0F0G0I000K0L0000M000P0R0S00
                           Y0
```

The status returned after receiving a U1 through U40 is an ASCII character 1 or 0, depending on the level of the line, and the state of the Invert (I16) command.

```
PRINT#1,"CLEAR08"          reset the Digital488/80A
PRINT#1,"OUTPUT08;U22X"    request the status of bit 22
INPUT#2,A$                 read the status
PRINT A$                   display shows a 0 (dependent on the signal
                           applied to the input)
```

Below is a summary of the Status (U0) information.

C#	Configuration
C0	All ports are inputs
C1	Port 1 is an output, ports 2 through 5 are inputs
C2	Ports 1 and 2 are outputs, ports 3 through 5 are inputs
C3	Ports 1 thru 3 are outputs, ports 4 and 5 are inputs
C4	Ports 1 thru 4 are outputs, port 5 is an input
C5	All ports are outputs

E#	Error Message
0	No error
1	Unrecognized command (ex. W3)
2	Invalid command option (ex. F8)
3	Conflict (ex., attempting to output data to an input port)
4	Not used
5	Non-Volatile RAM checksum error
6	Buffer Overrun error

F#	Data Format
F0	Hexadecimal
F1	ASCII
F2	Binary
F3	Decimal
F4	High Speed Binary

G#	Bus Input/Output mode
G1	Only Input port data sent when addressed to talk
G2	Only Output port data sent when addressed to talk
G3	Buffered input and output data sent when addressed to talk
G4	Continuous buffered input and output data sent when addressed to talk

I###	Invert Control Lines
I0	All control and data lines are active high
I1	Inhibit output is active low
I2	Trigger output is active low
I4	Data Strobe output is active low
I8	Clear output is active low
I16	Data are active low
I32	EDR input is falling edge sensitive
I64	Service input is falling edge sensitive

Note: The status indication reflects the sum of all received Invert commands.

K#	End Or Identify
K0	EOI enabled
K1	EOI disabled

L####	Buffer Status #### = number of readings in the buffer (0 to 2000)
M##	Service Request Mask
M0	SRQ is disabled
M1	SRQ on Service input transition
M2	SRQ on EDR input transition
M4	SRQ on Bus error
M16	SRQ on Ready

Note: The status indication reflects the sum of all received Service Request Mask commands

P#	Selected Port
P0	All ports selected
P1	Port 1 selected
P2	Port 2 selected
P3	Port 3 selected
P4	Port 4 selected
P5	Port 5 selected

R#	Data Ready
R0	Data are not latched, but are read when the Digital488/80A is addressed to talk
R1	Data are latched on EDR transition
R2	Data are latched and buffered on EDR transition
Y#	Terminator
Y0	CR LF
Y1	LF CR
Y2	CR only
Y3	LF only

View Configuration Vn

The View Configuration command is used to look at a configuration previously saved without loading it for use as the present configuration. In response to this command, the Digital488/80A returns a string in the following format:

S###C#F#G#I###K#M###P#R#Y#D00000000000Z

where each # equals the option for that command saved in the requested configuration. Data in between the D and Z are dependent on the state of the I/O lines at the time the configuration was saved.

Example:

```
PRINT#1,"CLEAR08"              reset the Digital488/80A
PRINT#1,"OUTPUT08;V12X"        view configuration 12
INPUT#2,A$,
PRINT A$                       display shows
                               S018C5F2G2I000K1M016P0R1Y2D000000
                               0000Z
```

This message is interpreted as follows:

S018	Configuration 18
C5	All ports are outputs
F2	ASCII Binary data format
G2	Only output port data sent when addressed to talk
I0	Data lines are active high
K1	EOI disabled
M16	SRQ on Ready
P0	All ports are selected
R1	Data are latched on EDR transition
Y2	Terminator is CR only
D00000000000Z	All I/O lines are low

View Version V?

The View Version command is used to look at the firmware revision level of the ROM. In response to this command, the Digital488/80A returns a number in the format `*.*` where each `*` is a one digit number. The Execute (X) command is not needed since View Version is a query command.

Example:

```
PRINT#1,"CLEAR08"          reset the Digital488/80A
PRINT#1,"OUTPUT08;V?"      request firmware revision level
PRINT#1,"ENTER08"          read data from the Digital488/80A
INPUT#2,A$,
PRINT A$                   display shows 1.0 (or higher)
```

Execute X

Commands sent to the Digital488/80A result in no action until the unit is instructed to execute these commands. This is done by sending an X, usually as the last character of a command string. Commands sent without an X are stored in an internal buffer until an X is received. Any number of Execute commands may be inserted into the same command string. Certain commands, such as Bit Set (An) require an X after each command in a string if more than one of that command is used within the same string.

Example:

`PRINT#1,"CLEAR08"`	reset the Digital488/80A
`PRINT#1,"OUTPUT08;F2"`	send F2 to the Digital488/80A command input buffer
`PRINT#1,"OUTPUT08;X"`	instruct the Digital488/80A to execute its command input buffer
`PRINT#1,"OUTPUT08;A1XA2X"`	two Bit Set (An) commands are within the same string

Terminator Yn

The IEEE 488 bus terminator defaults at power-on to the Terminator setting saved in configuration 0 (the power-on configuration). It also may be programmed for any combination of Carriage Return (CR) and Line Feed (LF). The Y0 mode is the most commonly accepted terminator (CR-LF). Y1 reverses the sequence to send LF-CR. Y2 sends CR only and Y3 sends LF only.

Y0	CR LF
Y1	LF CR
Y2	CR only
Y3	LF only
Y?	Return present Terminator selection

Example:

```
PRINT#1,"OUTPUT08;Y3X"          select line feed terminator
```

Note: To save as the new default setting, Terminator must be saved under the O0 (Recall/Save configuration) command.

Query ?

All Digital488/80A commands offer a Query (?) option which may be used to determine the present configuration or mode of a command previously executed. To use this option, the first letter of the command is used followed by a question mark (?). Any number of these query commands may be combined into one string to allow the user to construct a specialized status command requesting the Digital488/80A to return only that information which is of interest for a given application. The Execute (X) command is not needed when using the Query option.

? Return the present configuration or mode of the command preceding the ?

Example:

```
PRINT#1,"CLEAR08"          reset the Digital488/80A
PRINT#1,"OUTPUT08;M4X"     select SRQ on bus error
PRINT#1,"OUTPUT08;M?"      determine last M command executed
PRINT#1,"ENTER08"          read data from the Digital488/80A
INPUT#2,A$,
PRINT A$                   display shows M4
PRINT#1,"OUTPUT08;U0X"     request the status of Channel 1
PRINT#1,"ENTER08"          read data from the Digital488/80A
INPUT#2,A$,
PRINT A$                   display shows status message
                           *.*C#E#F#G#I###K#L####M###P#R#S##
                           Y#
```

Using the Query command to return the same status message:

```
PRINT#1,"OUTPUT08;V?C?E?F?G?I?K?L?M?P?R?S?Y?X"
                           request the status of Channel 1
PRINT#1,"ENTER08"          read data from the Digital488/80A
INPUT#2,A$,
PRINT A$                   display shows status message
                           *.*C#E#F#G#I###K#L####M###P#R#S##Y#F
                           255
```

Serial Poll Status Byte

The Serial Poll Status byte is sent upon receiving the serial poll (SPOLL) command from the controller. Refer to the Service Request Mask (Mn) command description for details on how the Serial Poll byte is affected. To enable each bit to reflect the true status of the device, the appropriate Mn command must be excecuted. Below is a description of the significance of each bit in the Serial Poll Status byte.

Bit Location	Value	(SRQ Bit Value if set to logic 1)
DIO1 (LSB)	1	Service Input transition
DIO2	2	EDR input transition
DIO3	4	Bus error
DIO4	8	Not used, always 0
DIO5	16	Ready for more commands
DIO6	32	Not used, always 0
DIO7	64	Service Request bit
DIO8 (MSB)	128	not assigned, always 0

Serial Poll Bit Description

DIO1 When enabled by the M1 command, DIO1 is set by a transition on the Service Input line (active transition state determined by the Invert (I64) command. DIO1 is cleared after the controller serial polls the Digital488/80A.

DIO2 When enabled by the M2 command, DIO2 is set on an EDR transition (active transition state determined by the Invert (I32) command). DIO2 is cleared after the controller serial polls the Digital488/80A.

DIO3 DIO3 is set when an invalid command is sent to the Digital488/80A. The M4 command enables a Service Request to occur when an invalid command is received. The bit is cleared after the controller sends a Status (U0) command and reads the status string from the Digital488/80A.

DIO4 DIO4 is not used, and is always a logic zero.

DIO5 The DIO5 bit is set after an entire command string has been received and processed by the Digital488/80A. The bit is cleared while the Digital488/80A is processing commands which have been received from the controller. When used with the M16 command, a Service Request is also generated when the DIO5 bit is set. An Execute (X) command must be received before the DIO5 bit can be cleared.

DIO6 DIO6 is not used, and is always a logic zero.

DIO7 When the Digital488/80A generates a Service Request, the DIO7 bit is set. This is used by the controller to determine that the Service Request was generated by the Digital488/80A.

DIO8 DIO8 is not used, and is always a logic zero.

Example:

`PRINT#1,"CLEAR08"`	reset the Digital488/80A
`PRINT#1,"OUTPUT08;M4X"`	select SRQ on Bus error
`PRINT#1,"OUTPUT08;F7X"`	send an invalid bus command. ERROR and SRQ indicator lights should illuminate
`PRINT#1,"SPOLL08"`	display should be 84 (64+16+4)

64 denotes the Digital488/80A was the source of the SRQ. 16 denotes the Digital488/80A is READY for more commands. 4 denotes a Bus error.

When serial polled, the SRQ indicator light turns off.

IEEE 488 Primer

5.1 History

The IEEE 488 bus is an instrumentation communication bus adopted by the Institute of Electrical and Electronic Engineers in 1975 and revised in 1978.

Prior to the adoption of this standard, most instrumentation manufacturers offered their own versions of computer interfaces. This placed the burden of system hardware design on the end user. If his application required the products of several different manufacturers, then he might need to design several different hardware and software interfaces. The popularity of the IEEE 488 interface (sometimes called the General Purpose Interface Bus or GPIB) is due to the total specification of the electrical and mechanical interface as well as the data transfer and control protocols. The use of the IEEE 488 standard has moved the responsibility of the user from design of the interface to design of the high level software that is specific to the measurement application.

5.2 General Structure

The main purpose of the GPIB is to transfer information between two or more devices. A device can either be an instrument or a computer. Before any information transfer can take place, it is first necessary to specify which will do the talking (send data) and which devices will be allowed to listen (receive data). The decision of who will talk and who will listen usually falls on the System Controller which is, at power on, the Active Controller.

The System Controller is similar to a committee chairman. On a well run committee, only one person may speak at a time and the chairman is responsible for recognizing members and allowing them to have their say. On the bus, the device which is recognized to speak is the Active Talker. There can only be one Talker at a time if the information transferred is to be clearly understood by all. The act of "giving the floor" to that device is called Addressing to Talk. If the committee chairman can not attend the meeting, or if other matters require his attention, he can appoint an acting chairman to take control of the proceedings. For the GPIB, this device becomes the Active Controller.

At a committee meeting, everyone present usually listens. This is not the case with the GPIB. The Active Controller selects which devices will listen and commands all other devices to ignore what is being transmitted. A device is instructed to listen by being Addressed to Listen. This device is then referred to as an Active Listener. Devices which are to ignore the data message are instructed to Unlisten.

The reason some devices are instructed to Unlisten is quite simple. Suppose a college instructor is presenting the day's lesson. Each student is told to raise their hand if the instructor has exceeded their ability to keep up while taking notes. If a hand is raised, the instructor stops his discussion to allow the slower students the time to catch up. In this way, the instructor is certain that each and every student receives all the information he is trying to present. Since there are a lot of students in the classroom, this exchange of information

can be very slow. In fact, the rate of information transfer is no faster than the rate at which the slowest note-taker can keep up. The instructor, though, may have a message for one particular student. The instructor tells the rest of the class to ignore this message (Unlisten) and tells it to that one student at a rate which he can understand. This information transfer can then happen much quicker, because it need not wait for the slowest student.

The GPIB transfers information in a similar way. This method of data transfer is called handshaking. More on this later.

For data transfer on the IEEE 488, the Active Controller must:

a. Unlisten all devices to protect against eavesdroppers.

b. Designate who will talk by addressing a device to talk.

c. Designate all the devices who are to listen by addressing those devices to listen.

d. Indicate to all devices that the data transfer can take place.

5.3 Send It To My Address

In the previous discussion, the terms Addressed to Talk and Addressed to Listen were used. These terms require some clarification.

The IEEE 488 standard permits up to 15 devices to be configured within one system. Each of these devices must have a unique address to avoid confusion. In a similar fashion, every building in town has a unique address to prevent one home from receiving another home's mail. Exactly how each device's address is set is specific to the product's manufacturer. Some are set by DIP switches in hardware, others by software. Consult the manufacturer's instructions to determine how to set the address.

Addresses are sent with universal (multiline) commands from the Active Controller. These commands include My Listen Address (MLA), My Talk Address (MTA), Talk Address Group (TAG), and Listen Address Group (LAG).

5.4 Bus Management Lines

Five hardware lines on the GPIB are used for bus management. Signals on these lines are often referred to as uniline (single line) commands. The signals are active low, i.e., a low voltage represents a logic "1" (asserted), and a high voltage represents a logic "0" (unasserted).

5.4.1 Attention (ATN)

ATN is one of the most important lines for bus management. If Attention is asserted, then the information contained on the data lines is to be interpreted as a multiline command. If it is not, then that information is to be interpreted as data for the Active Listeners. The Active Controller is the only bus device that has control of this line.

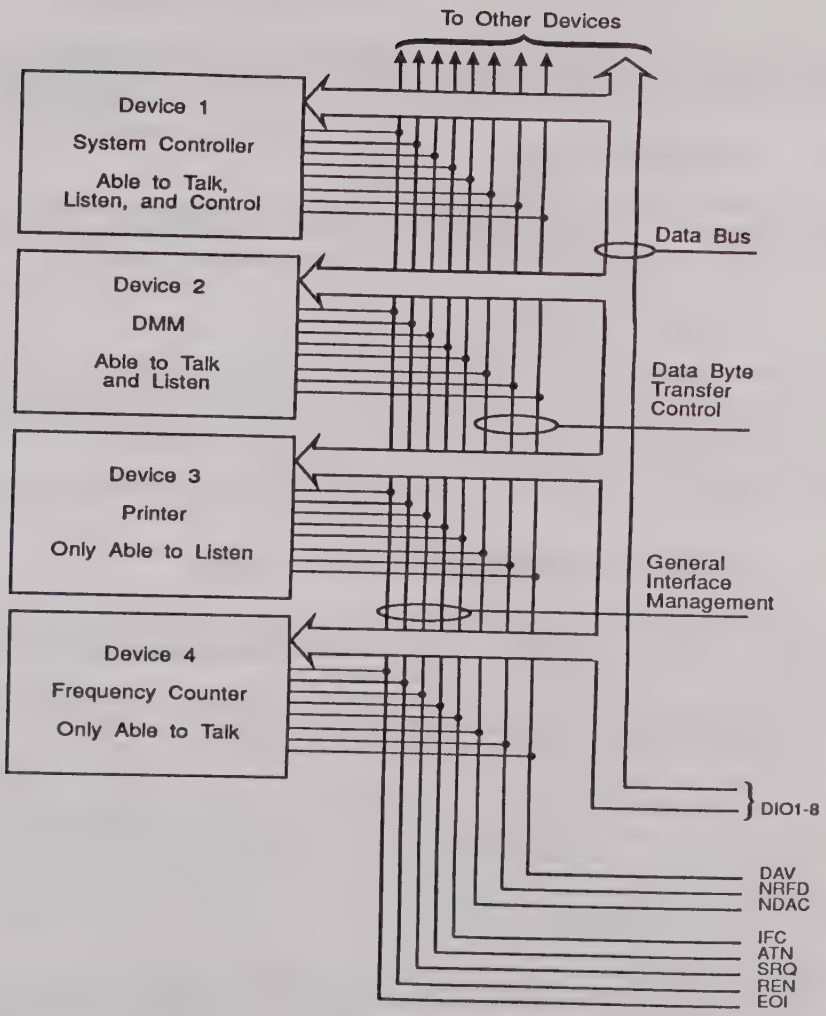


Figure 5.1: IEEE 488 Bus Structure

5.4.2 Interface Clear (IFC)

The IFC line is used only by the System Controller. It is used to place all bus devices in a known state. Although device configurations vary, the IFC command usually places the devices in the Talk and Listen Idle states (neither Active Talker nor Active Listener).

5.4.3 Remote Enable (REN)

When the System Controller sends the REN command, bus devices will respond to remote operation. Generally, the REN command should be issued before any bus programming is attempted. Only the System Controller has control of the Remote Enable line.

5.4.4 End or Identify (EOI)

The EOI line is used to signal the last byte of a multibyte data transfer. The device that is sending the data asserts EOI during the transfer of the last data byte. The EOI signal is not always necessary as the end of the data may be indicated by some special character such as carriage return.

The Active Controller also uses EOI to perform a Parallel Poll by simultaneously asserting EOI and ATN.

5.4.5 Service Request (SRQ)

When a device desires the immediate attention of the Active Controller it asserts SRQ. It is then the Controller's responsibility to determine which device requested service. This is accomplished with a Serial Poll or a Parallel Poll.

5.5 Handshake Lines

The GPIB uses three handshake lines in an "I'm ready - Here's the data - I've got it" sequence. This handshake protocol assures reliable data transfer, at the rate determined by the slowest Listener. One line is controlled by the Talker, while the other two are shared by all Active Listeners. The handshake lines, like the other IEEE 488 lines, are active low.

5.5.1 Data Valid (DAV)

The DAV line is controlled by the Talker. The Talker verifies that NDAC is asserted (active low) which indicates that all Listeners have accepted the previous data byte transferred. The Talker then outputs data on the bus and waits until NRFD is unasserted (high) which indicates that all Addressed Listeners are ready to accept the information. When NRFD and NDAC are in the proper state, the Talker asserts DAV (active low) to indicate that the data on the bus is valid.

5.5.2 Not Ready for Data (NRFD)

This line is used by the Listeners to inform the Talker when they are ready to accept new data. The Talker must wait for each Listener to unassert this line (high) which they will do at their own rate when they are ready for more data. This assures that all devices that are to accept the information are ready to receive it.

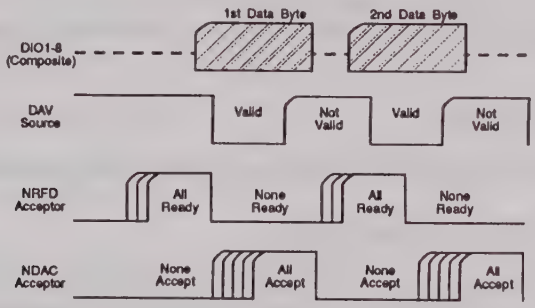


Figure 5.2: IEEE 488 Bus Handshaking

5.5.3 Not Data Accepted (NDAC)

The NDAC line is also controlled by the Listeners. This line indicates to the Talker that each device addressed to listen has accepted the information. Each device releases NDAC (high) at its own rate, but the NDAC will not go high until the slowest Listener has accepted the data byte.

5.6 Data Lines

The GPIB provides eight data lines for a bit parallel/byte serial data transfer. These eight data lines use the convention of DIO1 through DIO8 instead of the binary designation of D0 to D7. The data lines are bidirectional and are active low.

5.7 Multiline Commands

Multiline (bus) commands are sent by the Active Controller over the data bus with ATN asserted. These commands include addressing commands for talk, listen, Untalk and Unlisten.

5.7.1 Go To Local (GTL)

This command allows the selected devices to be manually controlled. ($01)

5.7.2 Listen Address Group (LAG)

There are 31 (0 to 30) listen addresses associated with this group. The 3 most significant bits of the data bus are set to 001 while the 5 least significant bits are the address of the device being told to listen.

5.7.3 Unlisten (UNL)

This command tells all bus devices to Unlisten. The same as Unaddressed to Listen. ($3F)

5.7.4 Talk Address Group (TAG)

There are 31 (0 to 30) talk addresses associated with this group. The 3 most significant bits of the data bus are set to 010 while the 5 least significant bits are the address of the device being told to talk.

5.7.5 Untalk (UNT)

This command tells bus devices to Untalk. The same as Unaddressed to Talk. ($5F)

5.7.6 Local Lockout (LLO)

Issuing the LLO command prevents manual control of the instrument's functions. ($11)

5.7.7 Device Clear (DCL)

This command causes all bus devices to be initialized to a pre-defined or power up state. ($14)

5.7.8 Selected Device Clear (SDC)

This causes a single device to be initialized to a pre-defined or power up state. ($04)

5.7.9 Serial Poll Disable (SPD)

The SPD command disables all devices from sending their Serial Poll status byte. ($19)

5.7.10 Serial Poll Enable (SPE)

A device which is Addressed to Talk will output its Serial Poll status byte after SPE is sent and ATN is unasserted. ($18)

5.7.11 Group Execute Trigger (GET)

This command usually signals a group of devices to begin executing a triggered action. This allows actions of different devices to begin simultaneously. ($08)

5.7.12 Take Control (TCT)

This command passes bus control responsibilities from the current Controller to another device which has the ability to control. ($09)

5.7.13 Secondary Command Group (SCG)

These are any one of the 32 possible commands (0 to 31) in this group. They must immediately follow a talk or listen address. ($60 to $7F)

5.7.14 Parallel Poll Configure (PPC)

This configures devices capable of performing a Parallel Poll as to which data bit they are to assert in response to a Parallel Poll. ($05)

5.7.15 Parallel Poll Unconfigure (PPU)

This disables all devices from responding to a Parallel Poll. ($15)

5.8 More on Service Requests

Most of the commands covered, both uniline and multiline, are the responsibility of the Active Controller to send and the bus devices to recognize. Most of these happen routinely by the interface and are totally transparent to the system programmer. Other commands are used directly by the user to provide optimum system control. Of the uniline commands, SRQ is very important to the test system and the software designer has easy access to this line by most devices. Service Request is the method by which a bus device can signal to the Controller that an event has occurred. It is similar to an interrupt in a microprocessor based system.

Most intelligent bus peripherals have the ability to assert SRQ. A DMM might assert it when its measurement is complete, if its input is overloaded or for any of an assortment of reasons. A power supply might SRQ if its output has current limited. This is a powerful

bus feature that removes the burden from the System Controller to periodically inquire, "Are you done yet?". Instead, the Controller says, "Do what I told you to do and let me know when you're done" or "Tell me when something is wrong."

Since SRQ is a single line command, there is no way for the Controller to determine which device requested the service without additional information. This information is provided by the multiline commands for Serial Poll and Parallel Poll.

5.8.1 Serial Poll

Suppose the Controller receives a service request. For this example, let's assume there are several devices which could assert SRQ. The Controller issues an SPE (Serial Poll enable) command to each device sequentially. If any device responds with DIO7 asserted it indicates to the Controller that it was the device that asserted SRQ. Often times the other bits will indicate why the device wanted service. This Serial Polling sequence, and any resulting action, is under control of the software designer.

5.8.2 Parallel Poll

The Parallel Poll is another way the Controller can determine which device requested service. It provides the who but not necessarily the why. When bus devices are configured for Parallel Poll, they are assigned one bit on the data bus for their response. By using the Status bit, the logic level of the response can be programmed to allow logical OR/AND conditions on one data line by more than one device. When SRQ is asserted, the Controller (under user's software) conducts a Parallel Poll. The Controller must then analyze the eight bits of data received to determine the source of the request. Once the source is determined, a Serial Poll might be used to determine the why.

Of the two polling types, the Serial Poll is the most popular due to its ability to determine the who and why. In addition, most devices support Serial Poll only.

Service Information

6.1 Factory Service

IOtech maintains a factory service center in Cleveland, Ohio. If problems are encountered in using the Digital488/80A, first telephone the factory. Many problems can be resolved by discussing the problems with our applications department. If the problem cannot be solved by this method, you will be instructed as to the proper return procedure.

6.2 Theory of Operation

The heart of the Digital488/80A is a 6809 microprocessor [U1] supported by 8K bytes of firmware EPROM [U2] and 32K bytes of battery-backed-up RAM [U3]. Interface to the IEEE 488 bus is accomplished by a 9914 interface chip [U13]. A 65B22 [U14] generates real-time interrupts for the firmware operating system and also drives the front panel LEDs.

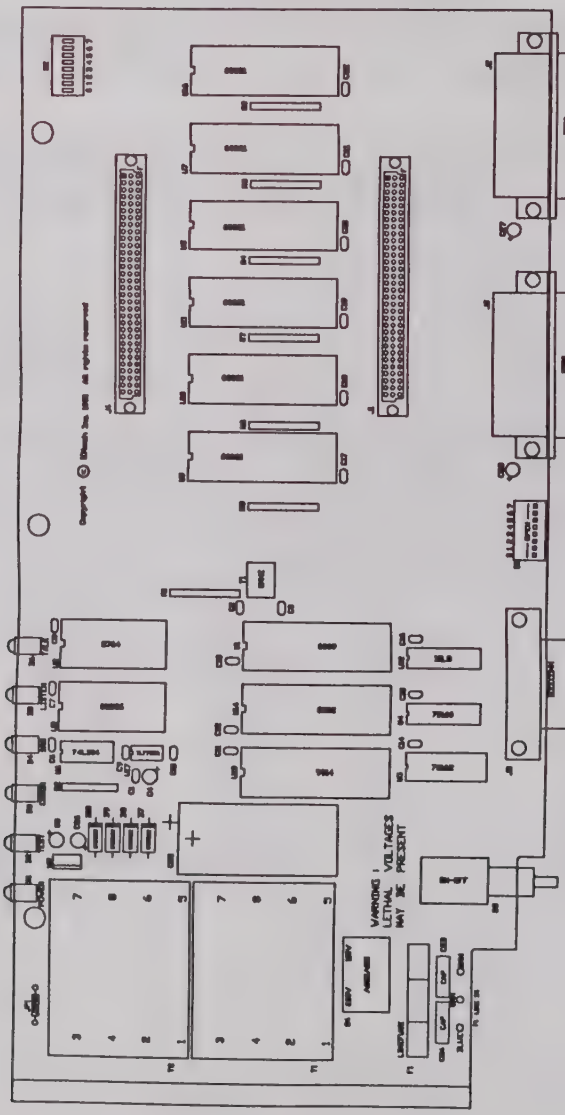
The digital I/O ports are controlled through 6821 interface chips [U7 - U11, U16]. High-voltage, high-current output capability is provided on both channels if the HVCX1 option is used.

Power is supplied by an on-board transformer and full-wave rectifier. Regulation to the required +5 volts is provided by an 78T05CT [U15].

The memory space allocation on the Digital488/80A is:

Address	Device	Part Number	Function
$0000-$7FFF	U3	84256	32K x 8 RAM
$8010-$8014	U7	68B21	Digital I/O
$8020-$8024	U8	68B21	Digital I/O
$8040-$8044	U9	68B21	Digital I/O
$8080-$8084	U10	68B21	Digital I/O
$8100-$8104	U11	68B21	Digital I/O
$8200-$8204	U16	68B21	Digital I/O
$8800-$8810	U14	65B22	Timer, Digital I/O
$9000-$9907	U13	TMS9914A	IEEE488 Interface
$E000-$FFFF	U2	2764	Programmed EPROM

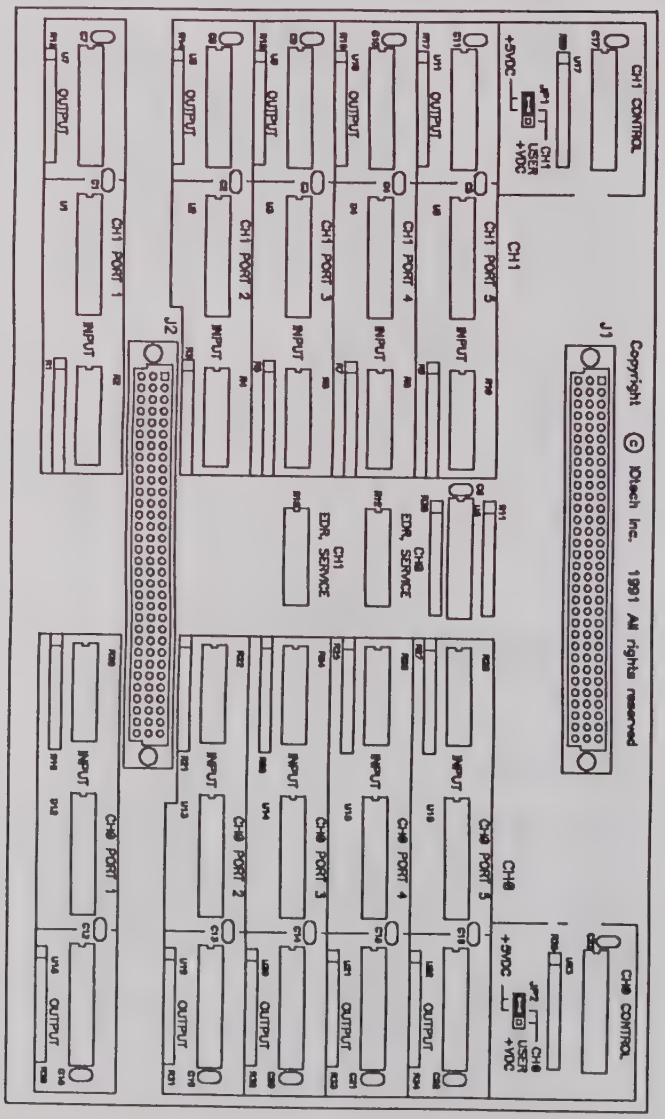
6.3 Digital488/80A Motherboard Component Layout



6.4 Digital488/80A Motherboard Replaceable Parts List

Schematic	Part Number	Description
C1, C6-C21	C-5-.1	0.1µF 25 V Ceramic Capacitor
C2, C3	C-5-15P	15pF 25 V Ceramic Capacitor
C4, C5	C-2-10	10µF 50 V Electrolytic Capacitor
C23, C24	C-11-4700P	4700pF 400 V Ceramic Capacitor
C25	C-6-12000	12,000µF Electrolytic Capacitor
C26	C-5-.33	0.33µF 25 V Ceramic Capacitor
C27, C28	C-2-10-63	10µF 63 V Electrolytic Capacitor
D1-D6	DD-2	LED
D7-D10	RF-4	3A Diode
F1	FU-1-.5	0.5 A Fuse
J2, J3	CN-8-50	DB50F Connector
R1, R3-R8	RN-9-10K	10K 9 Pin Resistor SIP Pack
R2	RN-4-470	470 ohm 7 Pin Resistor SIP Pack
S1	SW-1-8	8 Pole Piano DIP Switch
S2	SW-6-8	8 Position DIP Switch
S3	SW-3	DPDT Push-Push Switch
S4	SW-4	DPDT Slide Switch
T1, T2	TR-6	Transformer 10 V @1.2A
U1	IC-1	MC68B09P Microprocessor
U2	DIGIT488/80-600	Programmed EPROM
U3	IC-78	62256 Static RAM
U4	IC-4	75160 IEEE 488 Driver
U5	IC-5	75162 IEEE 488 Driver
U6	IC-33	74LS04 Hex Inverter
U7-U11, U16	IC-2	68B21 Peripheral Interface Adapter
U12	DIGIT488/80-601	Programmed PAL
U13	IC-3	9914 IEEE 488 Controller
U14	IC-23	65B22 Versatile Interface Adapter
U15	IC-165	78T05CT 3 A +5 V Regulator
U17	IC-108	TL7705A Power Failure Detector
Y1	CR-5	8MHz Crystal
J1, J4	CN-31-96	96-Pin DIN Connector (Female)

6.5 HVCX1 Option Component Layout



6.6 HVCX1 Option Replaceable Parts List

Schematic	Part Number	Description
C1 - C23	C-5-.1	0.1µF 25 V Ceramic Capacitor
R1, R3, R5, R7, R9, R11, R19, R21, R23, R25, R27	RN-9-15K	15K SIP Resistor Pack
R13, R14, R15, R16, R17, R29, R30, R31, R32, R33, R34, R35	RN-9-27K	27K Sip Resistor Pack
R2, R4, R6, R8, R10, R12, R18, R20, R22, R24, R26, R28	Depends on Application	See List in Section 2.2.2.2
U1 - U23	IC-205	2596 8-Channel Saturated Sink Drivers (See Section 2.2.2.2 for actual quantity required)
J1-J2	CN-31-96P	96-Pin DIN Connector (Male)

Appendix A: Digital 488/80A Command Summary

Bit Set	**An**	Set bit n (n = 1 thru 40) of selected channel (0 or 1)
	A?	Return n (1 thru 40) of last bit set of selected channel
Bit Clear	**Bn**	Clear bit n (1 thru 40) of selected channel (0 or 1)
	B?	Return n (1 thru 40) of last bit cleared of selected channel
Buffer	**L0**	Clear data buffer
	L?	Return number of readings in the data buffer
Bus Input/Output	**G0**	Input and Output port data sent when addressed to talk
	G1	Only Input port data sent when addressed to talk
	G2	Only Output port data sent when addressed to talk
	G3	Buffered input and output sent when addressed to talk
	G4	Continuous buffered input and output data sent when addressed to talk
	G?	Return present Bus Input/Output mode
Configure	**C0**	All ports of selected channel (0 or 1) are inputs
	C1	Port 1 is an output
	C2	Ports 1 and 2 are outputs
	C3	Ports 1 thru 3 are outputs
	C4	Ports 1 thru 4 are outputs
	C5	All ports of selected channel are outputs
	C?	Return present port configuration of selected channel
Data	**Dn...Z**	Data to be output on selected channel (0 or 1) is entered after "D" and terminated by "Z"

Data Ready	**R0**	Data is read when system is addressed to talk
	R1	Data is latched on EDR transition
	R2	Data is latched and buffered on EDR transition
	R?	Return present Data Ready mode
EOI	**K0**	EOI enabled
	K1	EOI disabled
	K?	Return present EOI selection
Error	**E?**	Return and clear present error condition
Execute	**X**	Execute preceding command string
Format	**F0**	ASCII Hexadecimal
	F1	ASCII Character
	F2	ASCII Binary
	F3	ASCII Decimal
	F4	Binary
	F5	High Speed Binary
	F?	Return present data format
Handshake	**H0**	Pulse the Clear line of selected channel (0 or 1)
	H1	Pulse the Strobe line of selected channel
	H2	Pulse the Trigger line of selected channel
	H?	Return last handshake command executed
Inhibit	**Q0**	Clear Inhibit line of selected channel (0 or 1)
	Q1	Set Inhibit line of selected channel
	Q?	Return last Inhibit command executed

Invert	**I0**	All control line outputs of selected channel (0 or 1) are active high data lines are high true
	I1	Inhibit output is active low
	I2	Trigger output is active low
	I4	Data Strobe output is active low
	I8	Clear output is active low
	I16	Data is low true
	I32	EDR input is falling-edge sensitive
	I64	Service input is falling-edge sensitive
	I?	Return last Invert command executed
Port	**P0**	All ports selected on channel m (m = 0 or 1)
	P1	Port 1 selected of channel m
	P2	Port 2 selected of channel m
	P3	Port 3 selected of channel m
	P4	Port 4 selected of channel m
	P5	Port 5 selected of channel m
	P?	Return present port selection of channel m
Query	**?**	Request command selection or device status
Recall Configuration	**On**	Load configuration n (n = 0 to 100) for use as present configuration
View Configuration	**Vn**	Return configuration n for examination without loading it for use as the current configuration
View Version	**V?**	Return revision level of Digital488/80 firmware
Save Configuration	**Sn**	Save present configuration as configuration n (n = 0 to 100)
	S?	Return last configuration (n) saved

SRQ Mask	**M0**	SRQ is disabled
	M1	SRQ on Service Input transition
	M2	SRQ on EDR input transition
	M4	SRQ on bus error
	M16	SRQ on Ready
	M?	Return present SRQ configuration
Status	**U0**	Send Status information of selected channel (0 or 1) when next addressed to talk (*.*C#E#F#G#I###K#L####M###P#R#S##Y#)
	Un	Read state of bit n (1 through 40) of selected channel
	U?	Return last Status command executed
Terminator	**Y0**	CR LF
	Y1	LF CR
	Y2	CR only
	Y3	LF only
	Y?	Return present Terminator mode
Test	**T0**	Turn off TEST indicator light
	T1	Turn on TEST indicator light

Appendix B: Keyboard Controller Program

BASIC Keyboard Controller Program

The Keyboard Controller Program is a simple BASIC program that accepts commands from the keyboard and sends them to Personal488. It then displays any responses from Personal488 on the screen. The Keyboard Controller Program is a convenient method of exercising the Digital488/80A and becoming familiar with the commands and their actions.

```
10 ' Personal488 Keyboard Controller Program
20 '
30 ' For use with the IOtech Personal488
40 ' IEEE 488 interface
50 '
100 OPEN "\DEV\IEEEOUT" FOR OUTPUT AS #1
110 IOCTL#1,"BREAK"
120 PRINT#1,"RESET"
130 OPEN "\DEV\IEEEIN" FOR INPUT AS #2
140 '
150 ON ERROR GOTO 300
160 PRINT#1,"ERROR OFF"
170 '
180 LINE INPUT "CMD ",CMD$
190 PRINT#1,CMD$
200 '
210 IF IOCTL$(2)"1" THEN 180
220 PRINT INPUT$(1,2);
230 GOTO 210
290 '
300 ' Error Handler
310 '
320 IOCTL#1,"BREAK"
330 PRINT#1,"STATUS"
340 INPUT#2,ST$
350 PRINT CHR$(7);"Error #";MID$(ST$,15,2);": ";MID$(ST$,27)
360 RESUME NEXT
```

Character Codes and IEEE Multiline Messages

$00 NUL 0	$10 DLE 16	$20 SP 32 00	$30 0 48 16	$40 @ 64 00	$50 P 80 16	$60 ` 96 SCG	$70 p 112 SCG
$01 SOH 1 GTL	$11 DC1 17 LLO	$21 ! 33 01	$31 1 49 17	$41 A 65 01	$51 Q 81 17	$61 a 97 SCG	$71 q 113 SCG
$02 STX 2	$12 DC2 18	$22 " 34 02	$32 2 50 18	$42 B 66 02	$52 R 82 18	$62 b 98 SCG	$72 r 114 SCG
$03 ETX 3	$13 DC3 19	$23 # 35 03	$33 3 51 19	$43 C 67 03	$53 S 83 19	$63 c 99 SCG	$73 s 115 SCG
$04 EOT 4 SDC	$14 DC4 20 DCL	$24 $ 36 04	$34 4 52 20	$44 D 68 04	$54 T 84 20	$64 d 100 SCG	$74 t 116 SCG
$05 ENQ 5 PPC	$15 NAK 21 PPU	$25 % 37 05	$35 5 53 21	$45 E 69 05	$55 U 85 21	$65 e 101 SCG	$75 u 117 SCG
$06 ACK 6	$16 SYN 22	$26 & 38 06	$36 6 54 22	$46 F 70 06	$56 V 86 22	$66 f 102 SCG	$76 v 118 SCG
$07 BEL 7	$17 ETB 23	$27 ' 39 07	$37 7 55 23	$47 G 71 07	$57 W 87 23	$67 g 103 SCG	$77 w 119 SCG
$08 BS 8 GET	$18 CAN 24 SPE	$28 (40 08	$38 8 56 24	$48 H 72 08	$58 X 88 24	$68 h 104 SCG	$78 x 120 SCG
$09 HT 9 TCT	$19 EM 25 SPD	$29) 41 09	$39 9 57 25	$49 I 73 09	$59 Y 89 25	$69 i 105 SCG	$79 y 121 SCG
$0A LF 10	$1A SUB 26	$2A * 42 10	$3A : 58 26	$4A J 74 10	$5A Z 90 26	$6A j 106 SCG	$7A z 122 SCG
$0B VT 11	$1B ESC 27	$2B + 43 11	$3B ; 59 27	$4B K 75 11	$5B [91 27	$6B k 107 SCG	$7B { 123 SCG
$0C FF 12	$1C FS 28	$2C , 44 12	$3C < 60 28	$4C L 76 12	$5C \ 92 28	$6C l 108 SCG	$7C | 124 SCG
$0D CR 13	$1D GS 29	$2D - 45 13	$3D = 61 29	$4D M 77 13	$5D] 93 29	$6D m 109 SCG	$7D } 125 SCG
$0E SO 14	$1E RS 30	$2E . 46 14	$3E > 62 30	$4E N 78 14	$5E ^ 94 30	$6E n 110 SCG	$7E ~ 126 SCG
$0F SI 15	$1F US 31	$2F / 47 15	$3F ? 63 UNL	$4F O 79 15	$5F 95 UNT _	$6F o 111 SCG	$7F DEL 127 SCG
ACG	**UCG**	**LAG**		**TAG**		**SCG**	

ACG = Addressed Command Group
UCG = Universal Command Group
LAG = Listen Address Group
TAG = Talk Address Group
SCG = Secondary Command Group

www.manualsplus.com
1032259